WearWrite: Crowd-Assisted Writing from Smartwatches

Michael Nebeling[1], **Alexandra To**[1], **Anhong Guo**[1], **Adrian A. de Freitas**[1],
Jaime Teevan[2], **Steven P. Dow**[1], **Jeffrey P. Bigham**[1]
[1] Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[2] Microsoft Research, Redmond, WA, USA
{ mnebelin, aato, anhongg, adefreit, spdow, jbigham }@cs.cmu.edu, teevan@microsoft.com

ABSTRACT
The physical constraints of smartwatches limit the range and complexity of tasks that can be completed. Despite interface improvements on smartwatches, the promise of enabling productive work remains largely unrealized. This paper presents *WearWrite*, a system that enables users to write documents from their smartwatches by leveraging a crowd to help translate their ideas into text. WearWrite users dictate tasks, respond to questions, and receive notifications of major edits on their watch. Using a dynamic task queue, the crowd receives tasks issued by the watch user and generic tasks from the system. In a week-long study with seven smartwatch users supported by approximately 29 crowd workers each, we validate that it is possible to manage the crowd writing process from a watch. Watch users captured new ideas as they came to mind and managed a crowd during spare moments while going about their daily routine. WearWrite represents a new approach to getting work done from wearables using the crowd.

Author Keywords
Smartwatches; Wearables; Crowdsourcing; Writing.

ACM Classification Keywords
H.5.m. Info. Interfaces and Presentation (e.g., HCI): Misc

INTRODUCTION
Smartwatches provide immediate access to information from anywhere, but their physical limitations make performing complex tasks difficult. As a result, users are rarely able to take advantage of spare moments to do useful work such as writing from their watches. It is, for example, hard to jot down a note if inspiration strikes while walking, contribute feedback on a draft while waiting for a bus, or proofread edits while waiting in line at a coffee shop. This paper describes how to bypass the existing limitations of watch-based content creation by using the watch as an interface to the crowd.

Prior work has directly improved watch-based interaction by augmenting the available hardware [13, 14, 40] and developing new input methods [6, 24, 29]. However, while this increases the range of possible interactions, limitations with input and output continue to inhibit people's ability to create new content. Touch-based text input from a watch remains much slower than it is from other types of devices, and text-entry alternatives like speech-to-text are error prone. Additionally, limited output on a watch makes it difficult for users to understand complex information and presents a challenge for interface designers who want to provide rich context.

We propose overcoming these limitations by using crowdsourcing. While using the crowd to complete complex tasks like writing is difficult, shepherding the crowd through the process by providing feedback along the way has been shown to result in higher-quality outcomes [10]. We hypothesized that a smartwatch could provide a sufficient and effective interface to orchestrate crowds to create new content, while crowdsourcing in turn could provide a mechanism to overcome limitations of the watch and enable a much wider range of smartwatch interactions than currently possible.

To study this, this paper presents *WearWrite*, a system that connects a smartwatch user as the domain expert of a particular piece of writing with a novice crowd of writers recruited on demand from Amazon Mechanical Turk. As shown in Figure 1, WearWrite consists of two key components:

Watch User Interface WearWrite provides the watch user with a lightweight notification-driven watch interface that allows the user to track and approve completed crowd work, issue new tasks, and respond to worker questions *via* built-in speech recognition or recorded audio. It employs a mixed-initiative approach to automatically complete simple actions on the user's behalf, only requiring approval for significant edits that can be previewed on the watch.

Crowd Worker Interface WearWrite provides crowd workers with desktop access to the document being written. It wraps the document to focus attention on open writing tasks while providing the full context of the document. Our system uses a dynamic task queue to prioritize the specific writing tasks issued and managed by the author, and fills the queue with generic writing tasks as needed.

In one-week deployments of WearWrite with seven smartwatch users and 205 crowd workers, we validated that it is possible to manage the crowd writing process from a watch. Participants worked on different types of writing projects, ranging from blog posts to research paper introductions, and made significant progress from initial outlines to first drafts. Participants particularly appreciated having access to the doc-

CHI'16, May 07–12, 2016, San Jose, CA, USA

ACM 978-1-4503-3362-7/16/05...$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858169

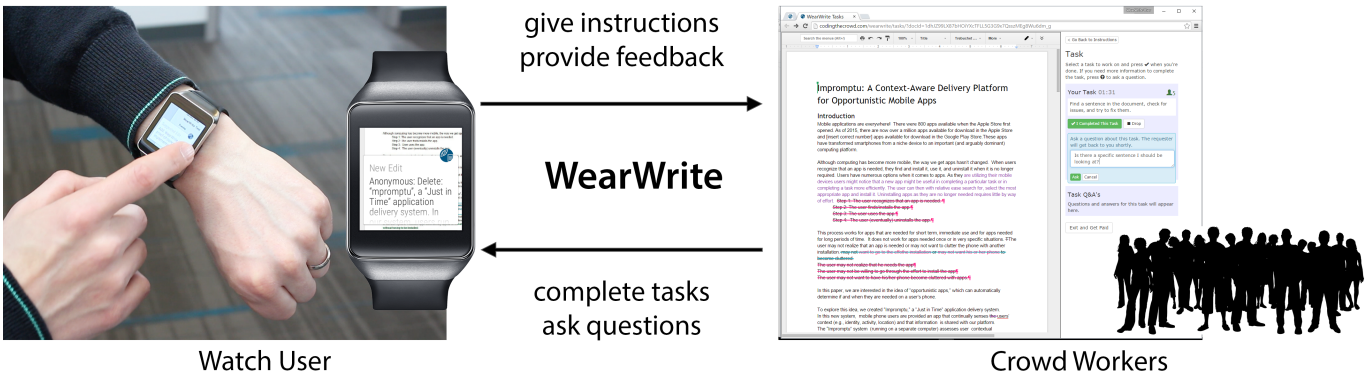

Figure 1. WearWrite lets users orchestrate complex writing tasks through their smartwatch. The smartwatch user provides a team of crowd writers with writing tasks and feedback. The crowd writers ask clarifying questions, work on a shared Google Doc, and deliver snippets of text for review.

ument while mobile so that they could quickly capture new ideas and offload writing tasks to crowd workers. WearWrite's lightweight watch interface allowed watch users to stay in the loop, but reviewing larger edits or getting context of many parallel tasks was still difficult. We discuss different strategies participants used for managing crowds from the watch on a range of writing tasks, the limitations of our current design, and insights for future work in this area.

RELATED WORK

WearWrite builds on prior work into *(i)* wearables and multi-device interaction, *(ii)* crowdsourcing of complex work, and *(iii)* crowd shepherding of collaborative writing.

Wearables and Multi-Device Interaction

Smartwatches are becoming increasingly popular, enabling new applications through glances and micro-interactions. A number of approaches have been introduced to increase the input capabilities of smartwatches and the amount of information that can be shown to a smartwatch user [2, 17, 22, 30, 32, 40]. Despite this, interaction on wearable devices like smartwatches remains limited. For example, text input from a smartwatch is much slower than it is from other types of devices [6, 29]. Speech-to-text is the primary technique used for smartwatch text input, but it can be error prone, especially for long sequences of text. It would be very cumbersome to write a document by typing or speaking long paragraphs of text using the smartwatch's existing technology.

There is promise in the research on multi-device interaction, which can allow a user to combine smartwatch interaction with larger devices to form a unified user interface and improve the range of possible interactions [25]. Researchers have started to explore how to develop systems and tools for building watch-centric, cross-device interactions [7, 9, 15]. However, the cross-device interfaces explored in this context so far are all designed to be used by a single user handling multiple devices. WearWrite brings in a different perspective on cross-device interfaces by allowing a user to provide input and interact with a document using their smartwatch on one end of the interface, and the crowd to perform actions on the user's behalf using larger and more powerful devices on the other end of the interface. Using this approach, WearWrite aims to overcome existing limitations and enable completing complex tasks on smartwatches.

By taking advantage of the spare moments users have during the day and allowing smartwatch users to recruit and orchestrate crowd workers, WearWrite expands the ways users can interact with documents. Related work suggests many potential advantages to helping people make use of short bursts of time while mobile [8]. There is evidence that information workers implicitly break larger tasks down into manageable subcomponents. People perceive tasks in segments [39], mental workloads dip at task boundaries [41], and many common tasks like email are already accomplished in short bursts [11]. The rising success of crowd work suggests traditional information workers stand to benefit from microwork structure [35], which can enable people to complete large tasks in many brief moments when they feel productive but do not have a long, uninterrupted period of time [4, 8, 37]. Additionally, providing people with the ability to complete productivity tasks while mobile in various different contexts has the potential to spark new perspectives on the same task [38].

Crowdsourcing Complex Work

Crowdsourcing is increasingly being used to complete complex work like writing. The most straightforward way to write with the crowd is to simply hire an expert writer *via* a site like UpWork. This is similar to what is currently done whenever writers share their work with a reader, editor, or collaborator. However, while expert-finding platforms reduce the friction of hiring an expert, there are still considerable cost and effort to working with a single individual. For this reason, crowdsourced creative tasks are often decomposed into smaller microtasks [19]. For instance, CrowdForge uses a partition-map-reduce pattern to provide a guide for decomposing complex tasks into context-free subtasks [20] and Turkomatic guides the crowd workers themselves to do the same using a price-divide-solve algorithm [21].

Crowdsourced writing is a particularly interesting domain in which to explore task decomposition and allocation because writing requires a number of fundamental but varied skills, and most traditional writing tools do not actively support the process of writing [12]. In recent years a number of different approaches have been tried to decompose the process of writing into microtasks [3, 16, 18, 20, 34]. For example, Soylent splits writing projects into stages and invites crowd workers to make suggestions, shorten, and proofread text [3].

The MicroWriter breaks the task of writing into three types of microtasks—generating ideas, labeling ideas to organize them, and writing paragraphs given a few related ideas—to produce a single report in short bursts [34]. Storia relies on the crowd to take in large amounts of information and generate written content summaries quickly [18]. The author can then leverage a more diverse set of content on the same information than they would have generated alone. WearWrite borrows task structure from this existing work, with the goal of having the crowd help the user write an article rather than having the crowd generate written content on its own.

Crowd Shepherding of Collaborative Writing

While task decomposition enables crowd workers to complete complex tasks like writing, the workers require oversight. Previous research has explored how requesters can best visualize crowd effort and provide feedback to the workers. The idea of "shepherding the crowd" [10] was introduced to help workers improve over time. Requester intervention during the work planning stage as well as reviewing and editing the crowd's work in real time significantly improves work quality [21]. CrowdForge builds in tools to easily insert quality control steps in a workflow to improve work quality [20]. For problems that require different kinds of expertise, teams of experts can be brought together on-the-fly to work together, as in Flash Teams [33]. Agapie *et al.* provide an example of this in the context of writing by using a combination of local and remote crowd workers to produce news articles [1]. Ensemble uses a team leader and an outline to direct crowd workers to ideate and contribute content [16]. In WearWrite, the watch user acts like the team leader in Ensemble.

WearWrite uses the crowd to allow authors to focus on the aspects of writing where they have unique insight to contribute. Collaborative writing is a common yet complex process that involves many discreet activities [31] and evolving roles [27]. Tomlinson *et al.* identified existing challenges with massively distributed collaborative writing, and found inadequate technological support for the process [36]. The WearWrite system attempts to fill this hole. Most collaborative writing currently relies on online synchronous collaborative authoring tools or the change tracking and version control features of modern word processors [28]. WearWrite adapts the best practices from collaborative writing by notifying users of important changes to the document. This paper contributes a system that allows a user to recruit, allocate tasks to, and provide feedback to crowd workers—all from a smartwatch interface. The watch provides a lightweight means of shepherding the crowd in a user's short bursts of spare time.

THE WEARWRITE SYSTEM

In this section, we present the WearWrite system, with a focus on its two user-facing components: the *watch interface*, which allows a smartwatch user to initiate new writing tasks and manage crowd work from their watch, and the *worker interface*, which is used by a crowd of writers recruited on demand to perform tasks requested by both the watch user and the WearWrite system. We describe the user experience with both interfaces first and then discuss the implementation.

Watch Interface

Figure 2 illustrates the three ways of interacting with crowd workers through WearWrite's watch interface. First, users can create new tasks for workers using two input methods: *(i)* the smartwatch's default speech-to-text interface or *(ii)* WearWrite's built-in audio recorder. Second, users can respond to questions submitted by workers. Watch users receive a notification on their smartwatch with the question and the corresponding task. They can then reply using the two input methods. In addition, they can cancel the task or create a new task from the notification. Third, while workers are working on a task, users will receive edit notifications together with a thumbnail of the document that highlights the edit and shows it in context. They can view the edit thumbnail in full screen on the watch and can accept or reject the edit from the notification. Below we provide a design rationale and describe the features of the watch interface in detail.

Creating Tasks

WearWrite supports creating tasks *via* the default speech recognizer or its own audio recorder. Speech-to-text may be preferred by watch users who achieve good accuracy with the recognizer, but can be problematic with longer instructions and those containing technical terms. This can make it necessary for users to repeat input multiple times until it is properly recognized. The audio recorder was developed to avoid recognition issues and create a user experience similar to that of a digital voice recorder. If the first option is used, workers will receive instructions in plain text. If the second option is used, an audio player is embedded in the worker interface that automatically plays the recorded instructions and workers can rewind and replay the audio as needed.

Accepting or Rejecting Edits

In WearWrite, all edits created by workers become suggestions. While this is required for the system to detect changes in the document, it can lead to confusion and frustration when many small edits are made in sequence and when many workers edit in parallel. Informing the watch user of every change was not an option, nor was automatically approving all changes without review. To reduce the load on the watch user, we developed a mixed-initiative approach that requires the watch user only approve major edits. The system automatically accepts minor edits without sending a notification to watch users. This approach supports both smartwatch users and crowd workers by reducing the chance for the document to quickly become messy with many small edits to the text.

WearWrite uses the following heuristics to distinguish between minor and major edits. Format changes are always minor and automatically accepted by the system. We define insert edits as minor if they are shorter than 60 characters and replace/delete edits if shorter than 30 characters. Longer edits are considered major edits and need to be approved by the watch user. These thresholds were determined in pilots with the system and have achieved good results in our deployments of WearWrite reported later. The barrier was set higher for automatic replace/delete edit acceptance to avoid significant portions of text being erroneously deleted by workers. As with format edits, we also considered to always automat-

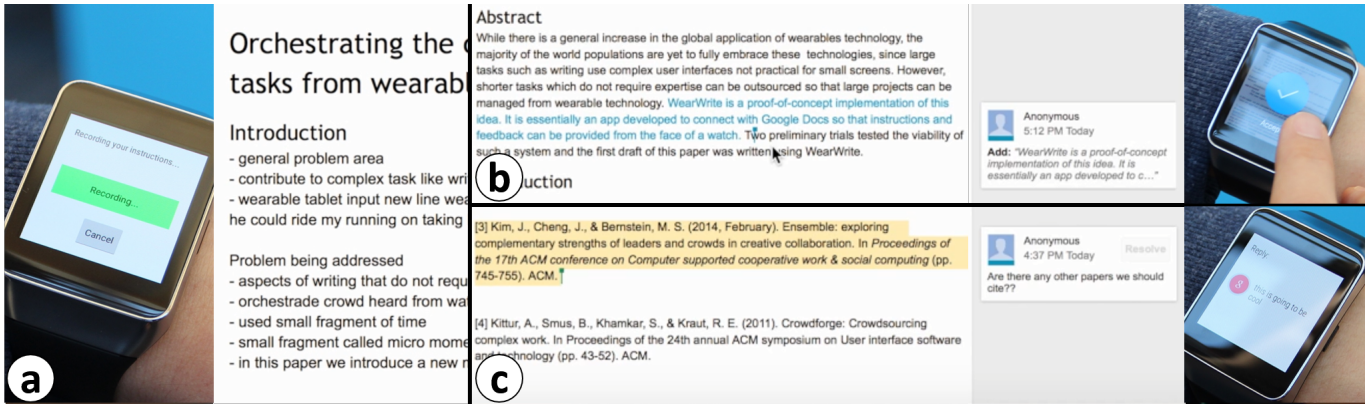

Figure 2. WearWrite's watch interface enables three main interactions: (a) create new tasks *via* speech-to-text or recorded audio instructions; b) review completed work and accept/reject edits based on document thumbnails showing them in the context in which they were made; c) answer questions on a task submitted by crowd workers, again using speech recognition or audio input.

ically accept insertions. However, to keep watch users in the loop, we decided to keep the threshold in order to notify them if longer sequences of text are added to the document.

Keeping Track of Work and Receiving Worker Feedback

In addition to edit thumbnails, watch users are also sent notifications when workers have completed tasks. We decided to send both edit and task notifications for two reasons. First, edit notifications inform a watch user about local changes to the document, whereas task notifications signal to users when major blocks of work have been completed. Second, task notifications allow users to keep track of worker progress in terms of their specific writing goals, prompting them to create new tasks building on the work accomplished so far.

In addition, users will receive comments that workers might choose to enter after they have performed a task or at the end of a session when they submit their work. While we had initially integrated this option for the evaluation of WearWrite's worker interface, our pilot studies showed that workers' post-task and post-work comments can provide valuable feedback on complexity and clarity of tasks, which we wanted to be sure to relay to the watch user requesting the task.

Viewing Tasks and Statistics on the Phone

The watch was designed to be the main interface for users. However, the system's core functionality is provided by the WearWrite mobile app installed on the smartphone that is paired with the watch. The phone app is responsible for processing watch user input, exchanging data with the WearWrite system and Google Docs, and sending notifications.

To focus the user's interactions on the watch in our experiments, we intentionally kept the phone interface minimalistic. In the current prototype of WearWrite, it lists tasks created by the user and shows statistics on tasks (*e.g.,* number of accepted, skipped and completed tasks by workers). In addition, it can also be used to accept/cancel tasks and play back audio tasks, which is useful for watches without audio output.

Worker Interface

Figure 3 shows the main page of the worker interface. The basic workflow for WearWrite workers is as follows. First, before viewing the document, they are given general instructions and information on compensation for tasks they complete and questions they ask. To achieve a quick turnaround and focus worker attention on the task, workers are advised to spend no more than five minutes per task. On the main page, workers read or listen to tasks given by the watch user depending on the input method used. They are free to skip and cycle through available tasks. Once they accept to work on a task, they will be allowed to edit the document. Workers can submit or drop the task at any time. They can work on as many tasks as they like during a session with WearWrite. If they have skipped tasks that were assigned to them, they will be reminded before they submit their work, and can choose to continue working on those tasks.

Dynamic Task Queue

At the core of the worker interface is a dynamic task queue. The system always fills this queue with generic writing tasks for workers, but pushes those specifically requested by the watch user to the top of the queue. As a result, workers will be assigned a watch user's specific tasks first ahead of any generic writing tasks. For workers who have previously worked on the document, this may be ideal. However, they also have the ability to skip tasks, allowing them to defer a task and resume it later. While certainly an option in the future, we opted against implementing locking on tasks so that only one worker will be able to accept a user's specific tasks. Rather, tasks can be accepted by multiple workers and will only be removed from the queue once the watch user cancels or accepts a task completed by a worker. To increase awareness when choosing a task, the number of workers that are currently working on that task is visible to every worker.

Generic Writing Tasks

The system generates writing tasks that are intentionally kept generic to potentially apply to many different types of writing. Our study also seeded four generic tasks: *(i)* "find a bullet point in the outline, and edit it so that it becomes a full sentence", *(ii)* "find a sentence in the document, check for issues, and try to fix them", *(iii)* "find a paragraph in the document, check the sentences, and try to improve them", and *(iv)* "skim through the document, find anything else that needs work, and improve it". These tasks were designed to guide workers

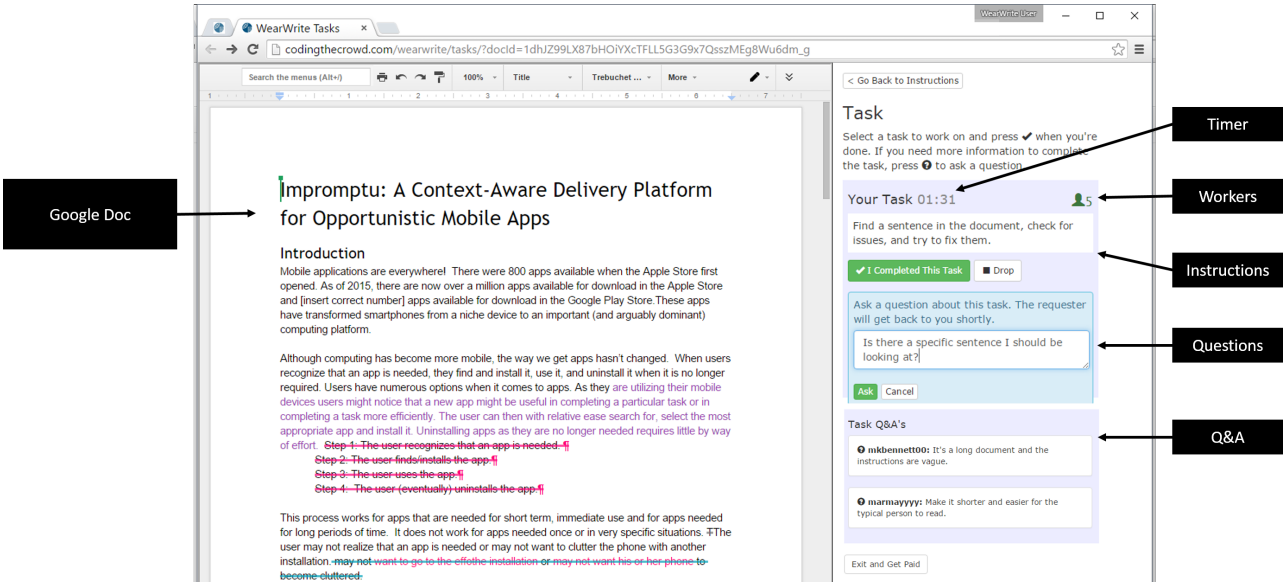

Figure 3. The worker interface wraps the Google Doc and makes it available for input once a task has been accepted. Next to the document are the task instructions in plain text or as an embedded audio playback control, a task completion timer, and the number of workers that have accepted the task. Below this is the Q&A interface for crowd workers to ask questions and see answers by the watch user related to the task.

through the process of starting from an outline and transforming it into prose, and continue iterating on the document from the level of sentences to increasingly larger portions of the document. The intention of this generic task model was that workers would gradually acquire more context of the writing by starting from tasks focusing on local edits and widening the scope of tasks on larger portions of the document.

Worker Feedback

Other than through the document itself, the worker interface provides three ways for workers to communicate with the watch user. First, they can submit feedback on task completion. Workers are asked to rate two statements, "the task was clear to me" and "I wanted more help on this task", on a 7-point Likert scale, and have the option of a free text response for any additional comments. Second, they can submit feedback at the end before they submit the HIT and return to Mechanical Turk. In this case, they are asked to rate three statements, "This is fair compensation for my work", "the tasks were interesting to me", and "the questions and answers on tasks were helpful", and enter a comment.

Questions & Answers

Finally, workers are always given the option to ask questions. The worker interface contains a specific area for questions and answers that will be visible to all workers working on the same task. Workers are incentivized to ask questions on the task by earning a small bonus for every question they submit. When adding this feature, we anticipated heavy use as it provides the primary means of two-way communication between workers and the watch user while they are working on a task. However, pilot testing revealed that workers hesitated to ask questions *during* tasks. Instead, they were much more likely to finish a writing task and leave a comment at the end. We therefore extended the worker interface to prompt workers to submit a question *post hoc* that they think, once answered by the watch user, would help improve and clarify the task.

Implementation

The implementation consists of three main components:

WearWrite App The app is divided into an Android Mobile app running on the phone and an Android Wear app running on the watch—the notification-driven watch interface allowed us to keep most of the logic on the phone;

WearWrite Server Implemented in PHP and responsible for scheduling tasks *via* the dynamic task queue and assigning them to workers—it hosts the worker interface and manages tasks, edits, and replies in a database and regularly sends updates on completed tasks, major edits, and worker questions to the Mobile app on the phone;

WearWrite Observer Implemented as a Chrome browser extension that we installed and hosted on a separate computer for our experiments, but could be implemented in a virtual browser such as PhantomJS and run within the WearWrite server—it periodically scans the Google Doc and extracts suggested edits, takes screenshots of major edits and sends them to the watch user, accepts or rejects edits as requested by the watch user, and automatically accepts minor edits without watch user approval.

WEARWRITE DEPLOYMENT

For the evaluation of WearWrite, we deployed the system with seven smartwatch users. Each participant used the WearWrite watch interface over the course of a week to create the first draft of a self-motivated writing project on a topic of their choice. We asked them to use it for an entire week in the context of their day-to-day activities so that they would have sufficient time and opportunity to explore WearWrite.

Participants

The seven participants (six male, one female, age 22-31 years) were recruited through university-wide mailing lists. To probe how WearWrite integrates with trained watch users'

Watch User						Crowd					Projects						
	Tasks		Major Edits				Work					Word Count			All Suggested Edits		
P#	Total	Audio	Total	Accept	Reject	Workers	Done	Time	Q's	Total Pay	Type	Outline	Draft	Diff	Total	Accept	Reject
1	11	29%	60	45%	2%	30	150	11.8 hrs	12	$44.54	Paper Intro	179	255	142%	81	95%	5%
2	10	100%	77	16%	0%	47	199	17.5 hrs	17	$69.51	Paper Intro	548	1012	185%	230	99%	1%
3	12	0%	30	0%	0%	18	63	3.9 hrs	6	$21.23	Blog Post	253	330	130%	68	99%	1%
4	18	0%	21	71%	5%	24	36	4.7 hrs	5	$10.17	Blog Post	87	203	233%	98	86%	14%
5	9	20%	34	12%	0%	26	44	3.9 hrs	17	$15.37	Paper Intro	243	358	147%	52	100%	0%
6	29	72%	26	0%	0%	29	42	5.0 hrs	12	$14.17	Blog Post	288	650	226%	198	99%	1%
7	15	9%	31	68%	3%	31	50	4.7 hrs	4	$15.87	Blog Post	85	633	745%	213	98%	2%
Total	**104**		**279**			**205**	**584**	**51.6 hrs**	**73**	**$190.85**					**940**		
Mean	**15**	**41%**	**40**	**30%**	**1%**	**29**	**83**	**7.4 hrs**	**10**	**$27.26**		**240**	**492**	**258%**	**134**	**96%**	**4%**

Figure 4. Statistics from our one-week deployments of WearWrite with seven participants, showing from left to right: the number of tasks and percentage of recorded audio instructions, the number of major edits sent to and percentage accepted/rejected on the watch, the number of workers, completed tasks and accumulated time on tasks, the number of questions submitted, the total money spent on crowd work, the type of project, word count at the beginning and end of the experiment with difference in percent, the number of all suggested edits and accepted/rejected in percent.

daily routines, participants were required to have owned and used a smartwatch for at least several months prior to the study. Actual use ranged from 3 months to a maximum of 3 years, with an average prior use time of 13 months. Two of the seven participants had previously worked with crowds from microtask platforms, but prior experience with crowdsourcing was not a requirement to participate in the study.

Study Protocol

The study was composed of three parts: *(i) Setup*—participants filled out a background questionnaire, installed the necessary software, and completed a short training after deployment, *(ii) Usage*—participants used the system as part of their daily life over the course of a week, and *(iii) Follow Up*—participants completed a post-study survey and exit interview. Participants were compensated with $55 USD.

Setup

In the Setup session, we installed WearWrite on the participants' personal smartphones and watches, and walked them through the process of creating tasks for workers and approving crowd work. Participants were also shown the worker interface so that they could observe how workers would receive new tasks and how workers could ask questions.

Participants were free in choosing the writing project they wanted to do using WearWrite. Since many of our participants were involved in research activities at the time of the experiment, three of them chose to write an introduction to a research paper they were already working on. The remaining four chose to write a blog post on a topic of their interest. To help jump start the writing with the crowd, each participant was asked to create a bulleted-list outline of their writing project on Google Docs.

Usage

Once the project outline was in place, the WearWrite Usage phase started. Participants were given a week to work with the crowd to evolve their initial outline into a prose first draft. A handout reminded smartwatch users that their online collaborators may be non-experts, and advised them to give short, specific, and actionable writing tasks, *e.g.,* "write two sentences on why a non-computer scientist would care about this work." Participants were asked but not required to use WearWrite regularly and as much as possible from the smartwatch. They always also had direct access to the Google Doc.

To always have a pool of workers available to participants, crowd workers were continuously recruited throughout the week. Across all projects a total of 205 crowd workers were hired from Mechanical Turk with an average of 29 workers per participant. For each project, WearWrite collected data on the participant's watch usage, including the number and types of tasks created, the number of edits accepted/rejected using the watch or Google Docs directly, and word count difference between the initial outline and the first draft produced at the end of the week. Google Docs kept a record of all revisions. WearWrite also collected data on how the crowd used the worker interface, including how often they skipped or completed tasks, the time spent, as well as their questions, post-task and post-work ratings and comments.

Follow Up

At the end of the study, we conducted a Follow Up session with each participant involving a 30 minute interview. For the first half of the interview, we asked participants to describe how their week progressed working with the crowd and the current state of their writing project. In the second half, we asked participants specifically about what they liked about the WearWrite system, and what they wished were different.

RESULTS

We start our presentation of results with an overview of each writing project of our seven participants. This is followed by an analysis of WearWrite usage statistics produced over the course of the week. Finally, we report feedback provided by our smartwatch users and crowd workers across all projects.

Overview of the Writing Projects

Figure 4 lists all projects completed by our smartwatch users with the statistics that WearWrite produced for the seven watch user participants and the 205 crowd workers.

P1 chose to write an introduction to a research paper on successful hackathon group traits. By using general crowd writing tasks to expand the bullet points from the outline provided by P1, the crowd quickly pushed towards a first draft. P1 then requested tasks asking to workers to merge certain sentences under the motivation into longer sentences and remove content that was already integrated elsewhere in the document. A final pass instructing crowd workers to consistently use the term 'hackathon' through the document produced a reasonable first draft for P1.

P2 wrote an introduction to a CHI submission he was working on describing a platform and technique to distribute mobile applications. Initially, P2 used a strategy similar to P1, first observing how crowd writers transformed the outline into an initial set of paragraphs. He then used primarily audio instructions, asking workers to elaborate on specific sentences, to add a transition between paragraphs, and finally to make the text "sound more formal, similar to something that one would read in a research paper". After he took a full pass over the document and made various edits on his desktop, he requested a shortening task using WearWrite, ending up with a complete introduction of a good length.

P3 wrote a chess tournament report that he wanted to publish on his blog. Compared to the writing projects of P1 and P2, this participant provided a more elaborate outline with pieces of sentences that he would like to see included in the document. The crowd was quick at connecting the different pieces and producing a first set of paragraphs. However, the document was not complete as P3 wanted to include the chess club's address, operating hours and tournament nights. While P3 had hoped that crowd workers would "Google this information by themselves," he finally provided the required details and formulated them as tasks for workers, which were then completed quickly.

P4 wrote on an amateur radio club and the activities and services they offer. She provided one of the shortest outlines and then filled workers in, creating tasks to add content to specific places in the document that should contain the information she provided. Compared to other participants, P4 managed small details more by first drawing the workers' attention to the subheadings that she wanted transformed into topic sentences, then asking them to reword certain sentences, merge individual sentences under subheadings into paragraphs, and remove redundancies, and finally having them improve the wording of specific sentences.

P5, like P2, used WearWrite to work on a CHI submission, in this case on smartwatch interactions. Again starting from bullets containing the basic arguments, workers quickly produced complete sentences and transitions between paragraphs. He then created an audio task to "make the sentences easier to read" which simplified some of the language, but also removed some technical terms that he had wanted to keep. He monitored the writing progress from the watch and also took an active role by editing the Google Doc from his desktop to bring back some of the details that were removed. Over the week, P5 increasingly used the Google Doc as a working document. He used it to fill in passages of text provided by one of his coauthors and asked crowd writers to correct the grammar and generally improve the English.

P6 wanted to write a blog post on 2016 US Presidential Candidates. He prepared an outline containing an introduction and then listing five potential candidates asking writers to write a short paragraph for each, providing their biographic details, party affiliations, previous work experience and views on crucial issues. He used unique strategy requesting a number of audio tasks at once aiming to get workers to focus on a specific candidate's profile. During the project he requested additional tasks to include specific aspects on some of the candidates and move some content around in the document.

P7 collected arguments on why Apple Inc. is successful. He provided a very simple outline containing five potential reasons. The document quickly grew to considerable length and contained various opinions of crowd writers. P7 then filtered and sorted some of the arguments, asking crowd writers to provide references to back up some of the claims. Like most projects, this document ended up with a draft following the initial structure, but putting the key arguments forward in prose. At the conclusion of all experiments, this project showed the largest difference in terms of word count comparing the initial outline and first full draft.

Usage Statistics

Figure 4 also shows the statistics we collected for the seven smartwatch users and the 205 crowd workers.

Watch Users

Over the course of a week, participating smart watch users created an average of 15 tasks for the workers per project. Usage of the option to record audio instructions varied a lot between participants from one using it all the time to two using it not at all. As the effect of our mixed-initiative approach, smartwatch users were only asked to approve 30% of all suggested edits and WearWrite automatically handled the rest for them. While there was a strong trend of accepting rather than rejecting edits, the tendency to accept edits directly from their watch varied considerably between participants. P3 and P6 approved all edits from their laptops rather than the watch, but they still kept abreast of changes *via* watch notifications.

Crowd Workers

Across all seven projects, 205 crowd workers were hired using Amazon's Mechanical Turk, with 142 unique worker ids. We initially experimented with different crowd platforms and piloted an early version of the system with expert writers hired from UpWork [26]. The expert workers paid a lot of attention to writing style and consistency and took a lot of initiative, at times even changing the examples desired by the watch user. We found that a larger crowd of non-expert workers reduces cost and makes it easier for the watch user to stay in control.

Crowd workers spent a total of 51.6 hours and made 940 edits. On average, participants worked with 29 crowd workers on their projects and workers spent 7.4 hours to complete 83 tasks per project, submitting 10 questions for each watch user. Workers worked for an hourly rate of $10 USD per HIT. We

developed a bonus system that paid up to $0.45 per task and a fixed $0.10 per question submitted. With 2:20 and 2:22 minutes on average, workers spent roughly the same time on system and user-generated tasks. All HITs paid regardless of edit approval, amounting to an average of $27.26 per project.

Workers transformed outlines into prose on the first day mostly with system tasks. After that, workers still improved the writing, but asked users for new tasks. While there was always worker activity, projects saw a burst of edits at the start and with new user tasks. On average, workers made a 134 edits per project with an average acceptance rate of 96% by our watch user participants. Most projects showed a substantial increase in terms of word count at the end, with an average of 258% across all projects, ranging from 130% for P3's blog post on a chess tournament, to 745% for P7's blog post on US 2016 presidential candidates. All projects started from outlines. P2 and P5 provided new text during the week. WearWrite's system tasks assume an existing outline or text. But P4's and P6's strategy to request a number of tasks in parallel allowed them to start with less content.

Qualitative Feedback from Watch Users

We now look at what we learned using the post-study questionnaires and interviews with the watch users. As indicated by the post-study questionnaires, three participants were convinced that WearWrite was useful for producing a first draft. The other participants did not express as much agreement for different reasons which we followed up with in the interviews. The watch interface was rated positively by most participants regarding ease of use, helpfulness for tracking progress and effectiveness in managing crowd work. Four of our seven participants argued that the crowd did not write similar to what they would produce, but most agreed that the questions and comments they received from crowd workers provided good feedback. Five of them wanted to continue using WearWrite in the future. Below we discuss six emerging themes from the interviews.

Potential to Transform Smartwatches

Before our study, several participants expressed apprehension with relying on their smartwatch for their writing project. However, afterwards they felt that WearWrite enabled productive work from their watches: "*I don't usually produce things on my smartwatch, it's only for review, so this was new*" (P1). "*I don't think you can do anything productive with the watch these days. I was surprised I could do something interesting*" (P3). Some even seemed to prefer WearWrite over other means of requesting help with writing: "*It's hard to give elaborate instructions for writing through email, the way I could quickly give a task was much better*" (P6).

Flexibility in Use

Smartwatch users were enthusiastic about the mobility the system provided: "*It was nice to see it progress while I was attempting to do something social*" (P4). Participants reported to have used WearWrite to create and review tasks in a variety of contexts. Most users successfully integrated the system into their daily routines. Five of seven participants used WearWrite in spare moments such as riding a bus, at a bus stop, waiting in line, at home, at work, and at a bar. On the other hand, two users scheduled specific times for WearWrite use: "*I would start in the morning, batch a bunch of tasks, and review all the work at the end of the day when I got home*" (P2).

Feeling of Productivity

Despite some concerns, participants saw potential in the crowd writing experience: "*having the crowd write stuff was pretty cool*" (P1). "*Having a framework where I can delegate tasks and scaffold different parts of my paper? I think that's useful*" (P5). Generally, they enjoyed offloading the writing tasks to the crowd so they could spend time working productively on other things. "*I don't feel so behind on my work, I know someone is taking care of this other project*" (P2). "*[The work] is not hard to do, but if someone else can do it, then that's really helpful to me*" (P5). One user even created and approved tasks while while at a board game night with friends and on a road trip, "*it's like an 8-hour drive home and I'm still able to make substantial edits*" (P4).

Easy to Request Tasks, Hard to Review

Regarding WearWrite's interface, watch users liked the low barrier to create tasks: "*I liked issuing tasks quickly from the watch*" (P2). "*I liked that I could create documents on the go, otherwise it's hard to use your laptop or phone while you're travelling or walking.*" (P3). Three watch users had a learning curve over the first few days, struggling to balance when and how many tasks to post. "*It was kind of a mess, several crowd workers would work on the same task so they would keep deleting work or sometimes there would be many redundancies*" (P1). In that situation, they often had to intervene and manually edit the document to restore sections or resolve conflicts. Two particularly struggled with reviewing large edits from the watch, "*if there was 1-2 sentences I could read it, but after that I always had to look at my phone or laptop to approve the edits*" (P7). When several tasks were requested in parallel, it was also hard to get the context of each completed task: "*when many tasks were completed at once, I couldn't get the context of where I was looking just from the watch... I started to do one task at a time so I always knew what I was reviewing and where it would be in the doc*" (P5).

Mixed Feelings about Quality

The most controversial issue in the interviews was writing quality. Those who felt positive about the experience argued that crowd workers successfully "*took an outline and turned it into prose, and I appreciate that*" (P4). One participant put it as follows: "*You are like the editor of a magazine. It was close, what they wrote was close to what I would write. There's not a huge difference.*" (P3).

However, many expressed apprehension about working with the crowd on writing projects. For example, P5 felt that, while the crowd was useful, there was an upper limit to what they could do. "*If it doesn't require my expertise, I value being able to have someone do that for me. [...] I write in a very particular style, so there's a mismatch, but I can always use the crowd as a starting point*" (P5). One participant who used WearWrite for writing an introduction to a CHI submission said "*I knew it wouldn't be good enough to copy and paste*

into my paper", but felt he could leverage the creative diversity of the crowd: "*This would be nice in the brainstorming phase; I like the idea of having lots of people help me come up with ideas*" (P2). For writing projects with more technical terms or jargon, a few users noted that the crowd would simplify by removing things they had wanted to keep. "*It's nice because it's readable... but it tends to be 'dumbing down' the text, the technical terms I wanted were deleted*" (P5).

Good Starting Point

All seven of the watch users planned to take the crowd's writing and use it for their final drafts of their projects. Four of the seven expressed that it was a good starting point, but that they will want to take an editing pass through the document. "*It's something to jump off of, and that's powerful and useful*" (P5). Three of the seven will take significant portions of the text to use directly without edits to the writing content. "*Some of the writers were really good... I didn't make any changes except for some formatting*" (P6). "*It's close to the final stage, I just want to add some figures*" (P3).

Qualitative Feedback from Crowd Workers

Through the worker interface, the crowd was able to provide feedback on WearWrite—both to the smartwatch users leading each project and to us as the designers of the system. We received 169 comments from workers in these ways, mostly related to WearWrite's worker interface and task design.

The post-task and post-work feedback showed a consistent theme in all projects across tasks. Workers rated tasks to be clear and interesting. The ratings also indicated that they did not want more help on tasks and that they found questions and answers helpful. Despite individual complaints, there was overall a strong agreement that compensation was fair. From the comments left by workers, we identified several common points of friction that they had while working on the projects.

Complex yet Interesting Interface

WearWrite's worker interface was received very positively by crowd workers. One worker provided a fairly comprehensive review: "*I haven't done an MTurk task like this thus far, and I liked that it had complexity, a clear and changeable setup, and the ability to give feedback and ask questions.*" To some workers, the WearWrite tasks also had a learning curve: "*There is a learning curve in this. It is an interesting project in that multiple people are working on it.*"

Jargon Blocks Productivity

Language was a common issue. Some of the projects had technical terms and jargon that made it difficult for crowd workers to contribute. "*I got hung up on the 'how it works' part because I realized that I just didn't have the technical knowledge to describe the step-by-step process.*" Even terminology around the writing process was sometimes a sticking point, *e.g.,* "*I don't know what a bullet point is.*"

Too Many Cooks

The seemingly most frustrating sticking point was the coordination issues that arose from multiple workers working simultaneously on the same thing. "*Its frustrating when others re-edit your work. Especially when they are terrible writers.*" "*Too many cooks in the kitchen spoils the broth.*" Some noted that their changes had been deleted (to them) prematurely.

Work Has Good Repeat Value

Overall, many workers expressed interest in working on similar projects. "*Very interesting to do more job like this please post the same and inform me to do so.*" Some crowd workers were engaged with the topic of their project, "*the subject matter is interesting.*" Some even felt that the work was personally fulfilling, "*it was a good learning experience. I hope to do this again, but with increased efficiency.*"

DISCUSSION

The goal of WearWrite is to leverage small moments of time to allow users to manage writing tasks on the go. WearWrite relies on a completely notification-driven interface – delivering document thumbnails to show edits in context, questions and comments by workers, and confirmations when work was completed. It uses a mixed-initiative approach that allows smartwatch users to make better use of their time by drawing their attention only to major document changes.

WearWrite's watch-centric design served as a probe to examine crowd-driven interfaces where the requester has limited interactive capabilities. While not all tasks can be completed from a watch, there are a class of tasks where using a watch rather than a phone might be less disruptive and more socially acceptable. Participants liked the challenge of mostly using the watch, but resorted to editing Google Docs on a laptop when they found the watch interface too limiting. WearWrite was useful for quick requests and feedback, and provided flexibility in use.

Judging from the feedback on WearWrite we received from both user groups, watch users and crowd workers, we can say that WearWrite takes a significant step forward, but our first prototype has not reached its full potential.

Better Supporting Transition between Devices

In our deployments of WearWrite, the current design pushed users to use the watch interface for almost all interactions with the system. This was considered crucial for our experiments to explore the benefits and limitations of writing from smartwatches using our approach. The feedback provided by participants indicates that users appreciated being able to initiate tasks from the watch. However, for other writing activities such as reviewing larger amounts of edits and to actively contribute to the writing, they preferred to use their mobile phone and desktop and so wanted to be able to more easily switch between devices as part of the writing process. This is something we can address by expanding on the mobile phone app that we have so far kept minimalistic, as well as improving the cross-device experience by adopting interaction techniques and design patterns from recent research [7, 25].

Interaction Between Users and Workers

We observed that our smartwatch users had a tendency to accept edits made by the crowd rather than reject them. We hypothesize that this could have happened for several reasons. For one, it may be that the crowd produced high quality edits that warranted acceptance. Users may also have a

bias towards accepting edits because any change, even when not clearly an improvement, made them feel like they were making progress. Users may also have been impacted by the fact that the edits were suggested by real people, and could have accepted changes in a desire to support the efforts for the Turkers. Previous work has shown that people interact with automated processes differently than with people, and it is interesting to consider the impact of exposing the humans on the other end of the watch on the watch user's experience.

There is also an open issue of how and when to recruit crowd workers for editing. For the experiments presented here, recruitment was done manually so that workers would be available during the times that the watch user were likely to be available. However, potential recruitment triggers include when the HITs run out, when a new task is set by the watch user, or at the request of the watch user. The service could also be scheduled so as to be available during a fixed set of hours, or as a function of predicted user availability.

Role of Workers in Authoring

In addition to providing insight into smartwatch interaction, WearWrite reveals several interesting things about collaborative writing from the perspective of the worker. While collaborative writing has been well studied [28, 31], crowd workers may represent a new type of collaborator. Further study is necessary to better understand how to best support this type of collaborator. For example, if workers interact with each other during writing, traditional collaborative editing tools may benefit their experience.

The copyright of textual content that is generated on commission belongs to the person who paid for it. However, when crowd workers contribute significant content to a document, it may be that they should be acknowledged as co-authors. It would be interesting to understand whether the 205 workers in our study felt like co-authors of the watch users' projects.

While our studies focused on crowd workers, the WearWrite workers do not have to be crowd workers. They could also be known collaborators or targeted experts in relevant domains (*e.g.,* in the topical domain of the piece being written, or in the domain of writing and copyediting), potentially pulled together on the fly [33]. The WearWrite workers could even be the same as the WearWrite smartwatch users themselves. Smartwatch users could use the worker interface to collaborate with themselves *via* the watch and worker interface [35].

Future Work

Future work in this area should investigate the trade-offs to farming out similar tasks and explore leveraging the crowd's expertise more thoroughly.

WearWrite's efficacy raises the question of how we should be spending our spare moments. What are the costs on our cognitive load to filling each free moment with a productive task? Do we lose value when we decompose projects into microtasks or does it allow us to focus better on the big picture?

While researchers have explored a number of different ways to structure writing [3, 16, 34], we looked at just one particular workflow for writing with WearWrite. It could be different structures lead to different performance. In particular, it may be able to design tasks that effectively transfer context-building steps to the crowd workers, *e.g.,* by better supporting the Q&A process. Recent work by Cai *et al.* [5] explores how doing chains of writing tasks creates context. Additionally, while WearWrite asked for feedback from the crowd after they completed tasks, it did not leverage the crowd in order to create tasks. Some crowd workers may take more initiative than others and a future version may encourage more collaborative efforts between the watch user and crowd workers.

We have explored the space of crowd writing, but are hopeful that WearWrite can be adapted to other creative or content-producing tasks. Future work on this topic may explore areas such as graphic or user interface design [23] from a watch.

CONCLUSION

This paper contributes the WearWrite system that enables users to write documents from their smartwatches by leveraging a crowd to complete writing tasks on their behalf. WearWrite users dictate tasks and receive notifications of major edits on their watch, while crowd workers work on both generic and user-generated writing tasks within a Google Doc. To explore this envisioned interaction, we evaluated WearWrite in week-long deployments with seven watch users. Participants appreciated WearWrite's flexibility and the increased productivity enabled by offloading writing tasks to the crowd. While the system allowed them to easily capture ideas throughout the day, it was still challenging to review large pieces of text. Crowd workers enjoyed the tasks and many came back to work on multiple WearWrite projects; however, some workers faced coordination issues and confusion over jargon. All seven authors went on to adapt versions of the crowd writing for their final drafts, with several using significant portions of the crowd's text without edits. WearWrite may be best used to complement, rather than replace, writing activities on a phone or laptop. The study provides a proof-of-concept that crowd-supported watch applications can provide a suitable approach for writing on the go.

ACKNOWLEDGMENTS

This research was supported by a Swiss National Science Foundation mobility grant, P300P2_154571, National Science Foundation grants, #1208382, #122206, and #1149709, and a Sloan Fellowship. We thank Kyle Murray for initial work on the watch interface.

REFERENCES

1. Elena Agapie, Jaime Teevan, and Andrés Monroy-Hernández. 2015. Crowdsourcing in the Field: A Case Study Using Local Crowds for Event Reporting. (2015), HCOMP '15.

2. Daniel Ashbrook, Kent Lyons, and Thad Starner. 2008. An Investigation into Round Touchscreen Wristwatch Interaction. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '08)*. ACM, New York, NY, USA, 311–314. http://dx.doi.org/10.1145/1409240.1409276

3. Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: A Word Processor with a Crowd Inside. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 313–322. http://dx.doi.org/10.1145/1866029.1866078
4. Carrie J. Cai, Philip J. Guo, James Glass, and Robert C. Miller. 2014. Wait-learning: Leveraging Conversational Dead Time for Second Language Education. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 2239–2244. http://dx.doi.org/10.1145/2559206.2581183
5. Carrie J. Cai, Shamsi T. Iqbal, and Jaime Teevan. 2016. Chain Reactions: The Impact of Order on Microtask Chains. In *Proceedings of the 34th Annual ACM Conference on Human Factors in Computing Systems (CHI'16)*. ACM, 6. http://dx.doi.org/10.1145/2858036.2858237
6. Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014a. Swipeboard: A Text Entry Technique for Ultra-small Interfaces That Supports Novice to Expert Transitions. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 615–620. http://dx.doi.org/10.1145/2642918.2647354
7. Xiang 'Anthony' Chen, Tovi Grossman, Daniel J. Wigdor, and George Fitzmaurice. 2014b. Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 159–168. http://dx.doi.org/10.1145/2556288.2556955
8. Justin Cheng, Jaime Teevan, Shamsi T. Iqbal, and Michael S. Bernstein. 2015. Break It Down: A Comparison of Macro- and Microtasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)*. 4061–4064. http://dx.doi.org/10.1145/2702123.2702146
9. Pei-Yu (Peggy) Chi and Yang Li. 2015. Weave: Scripting Cross-Device Wearable Interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3923–3932. http://dx.doi.org/10.1145/2702123.2702451
10. Steven Dow, Anand Pramod Kulkarni, Scott R. Klemmer, and Björn Hartmann. 2012. Shepherding the Crowd Yields Better Work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW'12)*. 1013–1022. http://dx.doi.org/10.1145/2145204.2145355
11. Victor M. González and Gloria Mark. 2004. "Constant, Constant, Multi-tasking Craziness": Managing Multiple Working Spheres. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 113–120. http://dx.doi.org/10.1145/985692.985707
12. Nick Greer, Jaime Teevan, and Shamsi T. Iqbal. 2016. *An introduction to technological support for writing*. Technical Report. Microsoft Research Tech Report MSR-TR-2016-001.
13. Chris Harrison and Scott E. Hudson. 2009. Abracadabra: Wireless, High-precision, and Unpowered Finger Input for Very Small Mobile Devices. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 121–124. http://dx.doi.org/10.1145/1622176.1622199
14. Chris Harrison and Scott E. Hudson. 2010. Minput: Enabling Interaction on Small Mobile Devices with High-precision, Low-cost, Multipoint Optical Tracking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1661–1664. http://dx.doi.org/10.1145/1753326.1753574
15. Steven Houben and Nicolai Marquardt. 2015. WatchConnect: A Toolkit for Prototyping Smartwatch-Centric Cross-Device Applications. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1247–1256. http://dx.doi.org/10.1145/2702123.2702215
16. Joy Kim, Justin Cheng, and Michael S. Bernstein. 2014. Ensemble: Exploring Complementary Strengths of Leaders and Crowds in Creative Collaboration. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 745–755. http://dx.doi.org/10.1145/2531602.2531638
17. Jungsoo Kim, Jiasheng He, K. Lyons, and T. Starner. 2007. The Gesture Watch: A Wireless Contact-free Gesture based Wrist Interface. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers*. 15–22. http://dx.doi.org/10.1109/ISWC.2007.4373770
18. Joy Kim and Andres Monroy-Hernandez. 2015. Storia: Summarizing Social Media Content based on Narrative Theory using Crowdsourcing. *arXiv preprint arXiv:1509.03026* (2015). http://arxiv.org/abs/1509.03026
19. Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The Future of Crowd Work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, New York, NY, USA, 1301–1318. http://dx.doi.org/10.1145/2441776.2441923

20. Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E. Kraut. 2011. CrowdForge: Crowdsourcing Complex Work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 43–52. http://dx.doi.org/10.1145/2047196.2047202

21. Anand Kulkarni, Matthew Can, and Björn Hartmann. 2012. Collaboratively Crowdsourcing Workflows with Turkomatic. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12)*. ACM, New York, NY, USA, 1003–1012. http://dx.doi.org/10.1145/2145204.2145354

22. Gierad Laput, Robert Xiao, Xiang 'Anthony' Chen, Scott E. Hudson, and Chris Harrison. 2014. Skin Buttons: Cheap, Small, Low-powered and Clickable Fixed-icon Laser Projectors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 389–394. http://dx.doi.org/10.1145/2642918.2647356

23. Walter S. Lasecki, Juho Kim, Nick Rafter, Onkur Sen, Jeffrey P. Bigham, and Michael S. Bernstein. 2015. Apparition: Crowdsourced User Interfaces That Come to Life As You Sketch Them. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1925–1934. http://dx.doi.org/10.1145/2702123.2702565

24. Luis A. Leiva, Alireza Sahami, Alejandro Catala, Niels Henze, and Albrecht Schmidt. 2015. Text Entry on Tiny QWERTY Soft Keyboards. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 669–678. http://dx.doi.org/10.1145/2702123.2702388

25. Michael Nebeling and Anind K. Dey. 2016. XDBrowser: User-Defined Cross-Device Web Page Designs. In *Proceedings of the 34th Annual ACM Conference on Human Factors in Computing Systems (CHI'16)*. http://dx.doi.org/10.1145/2858036.2858169

26. Michael Nebeling, Anhong Guo, Kyle I. Murray, Annika Tostengard, Angelos Giannopoulos, Martin Mihajlov, Steven Dow, Jaime Teevan, and Jeffrey P. Bigham. 2015. WearWrite: Orchestrating the Crowd to Complete Complex Tasks from Wearables (We Wrote This Paper on a Watch). *CoRR* abs/1508.02982 (2015). http://arxiv.org/abs/1508.02982

27. Christine M Neuwirth, David S Kaufer, Ravinder Chandhok, and James H Morris. 2001. Computer support for distributed collaborative writing: A coordination science perspective. *Coordination Theory and Collaboration Technology, ed. GM Olson, TW Malone, and JB Smith, Lawrence Erlbaum Associates, New Jersey* (2001).

28. Sylvie Noël and Jean-Marc Robert. 2004. Empirical Study on Collaborative Writing: What Do Co-authors Do, Use, and Like? *Computer Supported Cooperative Work (CSCW)* 13, 1 (2004), 63–89. http://dx.doi.org/10.1023/B:COSU.0000014876.96003.be

29. Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2799–2802. http://dx.doi.org/10.1145/2470654.2481387

30. Simon T. Perrault, Eric Lecolinet, James Eagan, and Yves Guiard. 2013. Watchit: Simple Gestures and Eyes-free Interaction for Wristwatches and Bracelets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1451–1460. http://dx.doi.org/10.1145/2470654.2466192

31. I.R. Posner and R.M. Baecker. 1992. How people write together [groupware]. In *System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on*, Vol. iv. 127–138 vol.4. http://dx.doi.org/10.1109/HICSS.1992.183420

32. M. T. Raghunath and Chandra Narayanaswami. 2002. User Interfaces for Applications on a Wrist Watch. *Personal Ubiquitous Comput.* 6, 1 (Jan. 2002), 17–30. http://dx.doi.org/10.1007/s007790200002

33. Daniela Retelny, Sébastien Robaszkiewicz, Alexandra To, Walter S. Lasecki, Jay Patel, Negar Rahmati, Tulsee Doshi, Melissa Valentine, and Michael S. Bernstein. 2014. Expert Crowdsourcing with Flash Teams. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 75–85. http://dx.doi.org/10.1145/2642918.2647409

34. Jaime Teevan, Shamsi T. Iqbal, and Curtis von Veh. 2016. Supporting Collaborative Writing with Microtasks. In *Proceedings of the 34th Annual ACM Conference on Human Factors in Computing Systems (CHI'16)*. ACM, 6. http://dx.doi.org/10.1145/2858036.2858108

35. Jaime Teevan, Daniel J. Liebling, and Walter S. Lasecki. 2014. Selfsourcing Personal Tasks. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 2527–2532. http://dx.doi.org/10.1145/2559206.2581181

36. Bill Tomlinson, Joel Ross, Paul Andre, Eric Baumer, Donald Patterson, Joseph Corneli, Martin Mahaux, Syavash Nobarany, Marco Lazzari, Birgit Penzenstadler, Andrew Torrance, David Callele, Gary Olson, Six Silberman, Marcus Stünder, Fabio Romancini Palamedi, Albert Ali Salah, Eric Morrill, Xavier Franch, Florian Floyd Mueller, Joseph 'Jofish' Kaye, Rebecca W. Black, Marisa L. Cohn, Patrick C. Shih, Johanna Brewer, Nitesh Goyal, Pirjo Näkki, Jeff Huang, Nilufar Baghaei, and Craig Saper. 2012. Massively

Distributed Authorship of Academic Papers. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems (CHI EA '12)*. ACM, New York, NY, USA, 11–20. http://dx.doi.org/10.1145/2212776.2212779

37. Rajan Vaish, Keith Wyngarden, Jingshu Chen, Brandon Cheung, and Michael S. Bernstein. 2014. Twitch Crowdsourcing: Crowd Contributions in Short Bursts of Time. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3645–3654. http://dx.doi.org/10.1145/2556288.2556996

38. Edward Vul and Harold Pashler. 2008. Measuring the crowd within probabilistic representations within individuals. *Psychological Science* 19, 7 (2008), 645–647. http://dx.doi.org/10.1111/j.1467-9280.2008.02136.x

39. Christopher D. Wickens. 2002. Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science* 3, 2 (2002), 159–177. http://dx.doi.org/10.1080/14639220210123806

40. Robert Xiao, Gierad Laput, and Chris Harrison. 2014. Expanding the Input Expressivity of Smartwatches with Mechanical Pan, Twist, Tilt and Click. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 193–196. http://dx.doi.org/10.1145/2556288.2557017

41. Jeffrey M Zacks, Barbara Tversky, and Gowri Iyer. 2001. Perceiving, remembering, and communicating structure in events. *Journal of Experimental Psychology: General* 130, 1 (2001), 29. http://dx.doi.org/10.1037/0096-3445.130.1.29