

MIT Open Access Articles

Learnersourcing Subgoal Labels for How-to Videos

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Weir, Sarah, Juho Kim, Krzysztof Z. Gajos, and Robert C. Miller. "Learnersourcing Subgoal Labels for How-to Videos." in 18th ACM Conference on Computer-Supported Cooperative Work and Social Computing, (CSCW 2015), March 14-18, 2015, Vancouver, BC, Canada.

As Published: <http://confer.csail.mit.edu/cscw2015/paper#!cscw432>

Publisher: Association for Computing Machinery (ACM)

Persistent URL: <http://hdl.handle.net/1721.1/94561>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



Learnersourcing Subgoal Labels for How-to Videos

Sarah Weir¹ Juho Kim¹ Krzysztof Z. Gajos² Robert C. Miller¹

¹MIT CSAIL
Cambridge, MA USA
{sweir, juhokim, rcm}@csail.mit.edu

²Harvard SEAS
Cambridge, MA USA
kgajos@eecs.harvard.edu

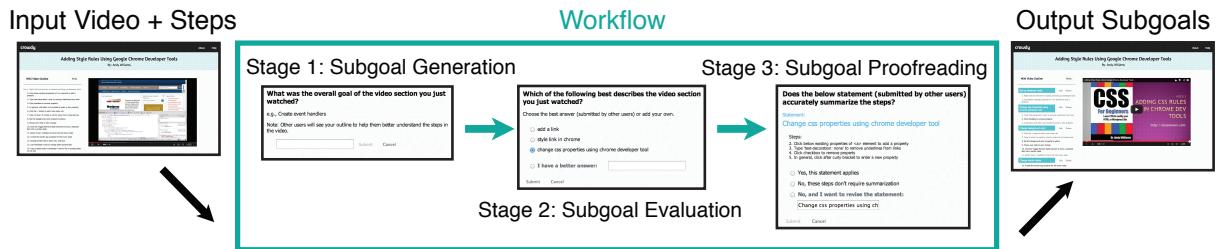


Figure 1. We present a three-stage workflow to generate labels for groups of steps (subgoal labels) for how-to videos. This workflow is designed to engage people actively trying to learn from the video to contribute the information.

ABSTRACT

Websites like YouTube host millions of how-to videos, but their interfaces are not optimized for learning. Previous research suggests that people learn more from how-to videos when the videos are accompanied by outlines showing individual steps and labels for groups of steps (subgoals). We envision an alternative video player where the steps and subgoals are displayed alongside the video. To generate this information for existing videos, we introduce *learnersourcing*, an approach in which intrinsically motivated learners contribute to a human computation workflow as they naturally go about learning from the videos. To demonstrate this method, we deployed a live website with a workflow for constructing subgoal labels implemented on a set of introductory web programming videos. For the four videos with the highest participation, we found that a majority of learner-generated subgoals were comparable in quality to expert-generated ones. Learners commented that the system helped them grasp the material, suggesting that our workflow did not detract from the learning experience.

Author Keywords

Video learning; subgoal labels; learnersourcing.

ACM Classification Keywords

H.5.1. Information Interfaces and Presentation (e.g., HCI): Multimedia Information Systems: Video.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CSCW '15, March 14 - 18 2015, Vancouver, BC, Canada
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2922-4/15/03\$15.00
<http://dx.doi.org/10.1145/2675133.2675219>

INTRODUCTION

More and more people are turning to the web to learn. While massive open online course (MOOC) platforms such as edX and Coursera have become popular for academic material, platforms like YouTube and Vimeo host millions of how-to videos that teach short procedural tasks ranging from putting on makeup to using Photoshop, to creating a two column layout using CSS.

Our long-term goal is to leverage research findings on how people learn and to find scalable approaches for improving the learning experience for viewers of existing how-to videos. In this work we specifically focus on the benefits conferred by augmenting instructional videos with textual outlines. Previous research [20] has shown that people learn better from videos if they are simultaneously presented with the *subgoals*, which capture meaningful conceptual pieces of the procedure covered in the video (Figure 2). Presenting subgoals improves learning by reducing learners' cognitive load by abstracting away low-level details, and by encouraging learners to self-explain why a set of steps have been grouped [4]. Ideally, to help people learn better from how-to videos, a video interface could display a list of subgoals as learners watch a video. Traditionally, labeling subgoals requires domain experts and knowledge extraction experts [5], but this process will not scale to the millions of existing how-to videos on the web. While makers of how-to videos may eventually be convinced to add subgoal labels for each video they create, we seek a scalable mechanism for improving the learning experience for viewers of existing videos.

In this paper, we turn to learners for help in generating subgoal labels at scale for how-to videos on the web (Figure 1). We introduce *learnersourcing*, a method for human computation for gathering information from people trying to actively learn from a video. Learnersourcing is a conceptual model in

which learners collectively generate useful content for future learners while engaging in a meaningful learning experience themselves. Unlike a paid crowd, learners are incentivized to watch these videos based on their own desire to learn, and not because they will receive payment. Learners may provide higher quality answers than a paid crowd because they are motivated by their interest in the instructional material. Like other crowd-powered systems, our system relies on a large number of micro-contributions and a computational mechanism to aggregate them. Unlike paid crowd workers, however, learners are unlikely to engage with tasks that appear tedious or irrelevant to their learning. A challenge in learnersourcing task design, therefore, is motivating learners to engage in the tasks while providing positive impact on learning experience or outcome. In other words, in learnersourcing we attempt to design meaningful human computation tasks while leveraging learners' natural activities.

While the idea of learnersourcing could have many applications, this paper presents a specific application for human computation in the educational video domain: subgoal labeling for how-to videos. Our solution is a three-stage workflow with microtasks for learners to complete as they watch the videos. We hypothesize that learners will be able to generate expert-quality subgoals with our workflow.

In the learnersourcing workflow we developed for constructing subgoal labels, as learners watch a video, the video stops periodically and the system asks one of three questions. The choice of question depends on how much information has already been gathered for that section in the video, and the questions are designed to engage learners to reflect on the content. In the first stage of the workflow, learners are asked to summarize the preceding short video section, which generates candidate subgoal labels for the section. After enough candidate subgoals have been generated for a video section, learners are routed to the second stage where they answer an automatically generated multiple choice question. The question asks them to pick the best subgoal from candidates contributed by previous learners. Learners' answers are then used by the system to evaluate the learner-generated subgoals. In the third stage, learners make improvements to the most popular subgoal by considering its scope and language. Once learners stop making changes to a subgoal, we can consider it final and future learners can be presented a final list of subgoals when watching the video.

To test our learnersourcing approach, we built Crowdy, a prototype system powered by the learnersourcing workflow. It features how-to videos on introductory web programming that demonstrate how to accomplish procedural tasks in HTML, CSS, and jQuery. We conducted a two-part deployment of Crowdy. The first was an in-class deployment at a university course on user interface design. We used results from that deployment to refine the interface and the workflow. We then conducted an in-the-wild study over a period of 25 days, where we made Crowdy publicly available. We found that in both deployments, learners were able to generate subgoals close to expert quality. In the live deployment, 14 out of 17 subgoals evaluated had at least one out of four evaluators

rate the learner subgoal as matching or better than the subgoal generated by experts. We interviewed three learners who emailed us to ask questions or express interest while using the site. We recognize that this sample is biased toward learners who used the site and is not representative of the entire population, especially those who dropped out. These learners felt the learnersourcing workflow helped them pay attention to the video and increased their understanding of the material. Based on these findings, we believe that learnersourcing is a viable solution for generating subgoal labels for how-to videos.

The contributions of this paper are as follows:

- An introduction of learnersourcing, a context-specific form of crowdsourcing in which learners engage in human computation tasks while trying to learn a new skill.
- An implementation of a three-stage learnersourcing workflow to generate high-quality subgoal labels from how-to videos in a scalable way.
- Results from an in-the-wild study demonstrating the workflow's effectiveness in generating high-quality subgoals. 14 out of 17 subgoals had at least one out of four evaluators rate the learner-generated subgoal label as equal to or better than the expert-generated label.
- Results from interviews with learners who participated in the in-the-wild study suggesting that the system helped them pay attention to videos and grasp the material better.

RELATED WORK

Previous work suggests that presenting learners with subgoals for procedural tasks improves learning. Eiriksdottir and Catrambone [8] explored the effects of presenting learners with different forms of instructional material. They discovered that including specific instructions helped learners complete the initial task but those learners did not retain the information. Conversely, learners presented with more holistic instructions had greater learning and transfer. Margulieux et al. showed that instructions including both specific steps and subgoals [20] resulted in improved learning and transfer, compared to those with the specific steps alone. Buykx and Petrie [3] included subgoals in recipe instructions and showed that displaying steps and subgoal information improves understanding in domains other than software applications. The TAPS method [5] proposes a systematic workflow for extracting subgoals for existing lecture videos with a domain expert and a knowledge extraction expert working together. This work suggests that knowledge extraction experts, who are novices in their domains, are a viable source for providing information to help other domain novices.

A rich body of previous research has looked into tutorial interfaces for learning software applications. Some of these interfaces focus on ways to visualize and discover the most suitable tutorials [17, 23], while other systems enhance tutorial content by building on the metadata available in software applications [6, 9, 16, 24]. The success of these systems in helping people learn to use software suggests that using existing information (such as the transcript to a tutorial video)

Overall goal

How to Make a Cake

Subgoal

Combine the dry ingredients

1. Put 2 cups of water into a mixing bowl
2. Add 1 cup of sugar to the bowl
3. Add 2 tbsp of baking soda
4. Add 1/2 tsp of salt
5. Stir together

Individual steps

Subgoal

Separately combine wet ingredients

6. In another bowl, beat two eggs
7. Add 1 stick of butter and beat
8. Add 1 cup of milk and stir

Individual steps

Figure 2. An example of the breakdown between goal, subgoals, and individual steps for a procedural task. Creating subgoals from individual steps requires self-explanation and summarization.

is an effective way to improve learning. However, many of these systems rely on the underlying structure of software applications to generate content [9, 17, 23, 24] and cannot easily be generalized to domains other than software.

Systems have also been built to leverage the power of the crowd to generate information on educational videos. For example, VidWiki [7] is a platform to crowdsource incremental improvements to educational videos with annotations. Additionally, ToolScape [16] used a Mechanical Turk crowd to generate step-by-step annotations for how-to videos. The fact that these systems were successful suggests that crowdsourcing is an effective way to create additional content for educational videos.

A number of prior crowd-powered systems [1, 16, 19, 22] use multi-stage workflows to generate expert-quality material in a variety of domains. Our three-stage workflow is inspired by previous research, but is designed explicitly for a crowd of learners who are naturally watching the video with their own desire to learn. Designing human computation tasks for voluntary learners presents a unique set of challenges, which we address with our workflow.

LEARNERSOURCING SUBGOAL LABELS

In order to generate subgoal labels for how-to videos, we use a learnersourcing approach. Our design goals are:

- To create a method for generating subgoal labels that does not require experts,
- To enable learners to generate expert-quality subgoals collaboratively,
- To design human computation tasks that are engaging and do not detract from the learning experience.

Due to the variety in video domain and presentation style on the web, an automatic approach to generating subgoals does not seem feasible. For example, some videos include text, some are silent and only show a person completing certain actions, and others rely on video and sound for demonstration. Automatically adapting to the various presentation and

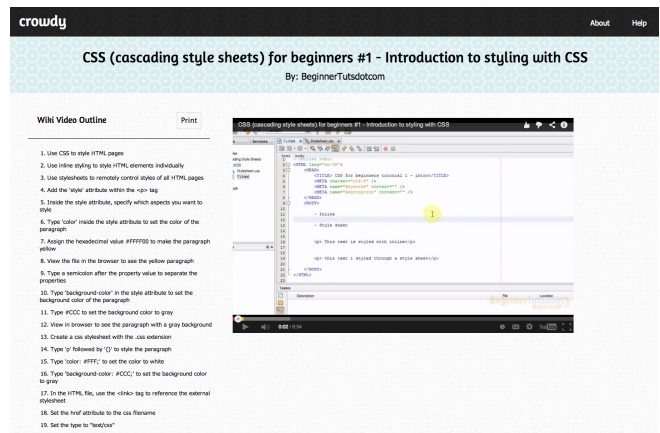


Figure 3. For each video on Crowdy, we show the video player as well as a side panel with the individual steps. After learners answer a question, the subgoal they added or picked appears in the outline. Learners can click on steps or subgoals to jump to various parts in the video. The question block (Figure 4, 5, 6) temporarily replaces the video player when it appears.

instructional styles is difficult, and we are looking for a solution that can generalize to any how-to video on the web. Additionally, subgoals are often at a higher conceptual level than what the video explicitly mentions, so automatically generating subgoals would require interpreting and understanding the material.

Learnersourcing is a context-specific form of crowdsourcing, in which people who are already trying to learn from a video are asked to complete microtasks while watching the video. Because learners are self-selected based on their interest in the video material, this approach is scalable and generalizable to any video domain. As long as there are learners interested in a video, there is the possibility for input into the system. Learnersourcing complements 'expertsourcing' (domain experts generating subgoal labels) and 'authorsourcing' (video authors generating subgoal labels) because it is designed to add subgoals to videos that already exist. Although not covered in this paper, well-designed learnersourcing prompts may offer a pedagogical benefit to learners.

We built a website, Crowdy, to implement our workflow for learnersourcing subgoal labels. It is an alternative site to watch videos that were originally hosted on YouTube. The front page features all videos for the current deployment. When a user selects a video, Crowdy shows a separate page with an embedded video player and interactive outline (Figure 3). Our learnersourcing workflow asks specific questions as learners watch the videos, and in turn generates high-quality subgoals for them. Crowdy was created using standard web technologies (HTML, CSS, jQuery), with a Django backend and the YouTube API (<https://developers.google.com/youtube/>) for the video player.

WORKFLOW DESIGN

In order to enable learners to generate subgoal labels from how-to videos, we designed a three-stage workflow that engages learners to create and refine each others' subgoal labels.

What was the overall goal of the video section you just watched?

e.g., Create event handlers

Note: Other users will see your outline to help them better understand the steps in the video.

Figure 4. In Stage 1, learners are asked to generate a new subgoal label after watching a video segment.

Initially, learners are presented with a video player and an interactive outline panel seeded with the low level steps that the video goes through (Figure 3). By aggregating the micro-contributions from learners, Crowdly adds subgoal labels to this outline. The steps are presented as a vertical timeline, such that clicking on a step moves the playhead to that time in the video. This research assumes that the low-level steps have already been generated for a video, and focuses only on generating subgoal labels. Although we manually added the steps for this research, individual steps could be obtained by crowdsourcing [16] or automatically in part, possibly by processing a transcript or notes from the video creators.

As learners watch a video, they are stopped periodically and asked a question about the preceding video section. Although we experimented with having a Wiki-like interface where learners were asked to contribute subgoal labels but were not forced to, we found that this direct question approach resulted in more frequent and higher quality participation.

We chose a multi-stage workflow because we wanted to make each task small enough so that learners could complete it without getting too distracted from the video. To ensure a high quality of final subgoal labels, we use three mechanisms used in prior crowd-powered systems: we solicit multiple candidate solutions, we use voting to identify the best candidates, and we allow for iterative refinement.

We automatically route each subgoal to the necessary stage, so a learner might experience different stage prompts even while watching a single video. By removing the need for human intervention and judgment to move to the next stage, we aim to standardize and improve the efficiency of the workflow.

STAGE 1: Generation

The goal of the first stage (Figure 4) is to have learners generate candidate subgoal labels from scratch. After a certain interval, learners are asked to summarize what they have just watched. While the learner is answering the question, the individual steps that were covered in the most recent video segment are bolded in the outline panel. Learners are given the option to cancel out of the question and are not forced to submit an answer.

In order to minimize spam answers, we made sure learners could easily cancel out of the prompt. We also designed the question such that it does not require prior familiarity with the concept of subgoals, but such that the answer to the question can serve as a subgoal label. From our preliminary user tests,

Which of the following best describes the video section you just watched?

Choose the best answer (submitted by other users) or add your own.

- importance of css, comparison to html
- defining css
- learn css

I have a better answer:

Figure 5. In Stage 2, learners vote on a subgoal they find most accurate.

we found that learners preferred jargon-free questions (i.e., using ‘goal’ instead of ‘subgoal’).

When there have been three candidate subgoal labels generated from the first stage, the next set of learners are asked the stage 2 question.

STAGE 2: Evaluation

The subgoal labels that learners generate in stage 1 are not always of the highest quality. Learners may interpret the question in different ways or have different levels of familiarity with the video’s topic, which cause their subgoal labels to be different from each other. In stage 2 (Figure 5), learners are presented with the previously generated subgoals and asked to choose which they feel best captures the preceding section.

The goal of the second stage is to choose the most relevant answer for the video section, as well as weed out potential spam answers. The input to the question in this stage are the subgoal labels previously generated by learners. The learners who are presented with the stage 2 question have the option to choose one of the subgoals, add their own, or abstain. We do not give learners any indication as to which subgoal is most popular during this stage. The subgoal that the learner chooses gets an upvote, and the ones that were not chosen receive a downvote. We use the upvote and downvote measures to determine when to move on to the third stage. This helps interpret user actions by weeding out answers that are consistently not chosen and considering newly generated subgoals. When the number of upvotes or the difference in upvotes between the top two subgoals reaches a certain threshold, we accept that subgoal as the winner of stage 2 and route the question to stage 3 for future learners.

STAGE 3: Proofreading

Learners in stage 2 evaluate the previously generated subgoal labels, but the most popular subgoal label from stage 2 is not necessarily a high-quality label by our standards. We define high-quality subgoal labels as having the correct scope (the subgoal label accurately reflects the underlying steps) and being concrete and descriptive. For example, for a section in a CSS video about changing the color of a div, a label like “CSS video” would be too broad, whereas a label such as “Use CSS to change div color” would be more appropriate.

The goal of this stage (Figure 6) is for learners to evaluate the most popular subgoal label from stage 2 for quality and eventually agree on a final label for that subgoal. The input to this

Does the below statement (submitted by other users) accurately summarize the steps?

Tip: Good statements usually begin with a verb.

Statement:

basic steps to creating an HTML template

Steps:

1. Begin HTML page with <html> tag and end with </html>
2. Start the <body> tag above content, end </body> below content
3. Create head section using <head> tag
4. Use the <title> tag to create page title

Yes, this statement applies
 No, these steps don't require summarization
 No, and I want to revise the statement:

basic steps to creating an HTML

Submit Cancel

Figure 6. In Stage 3, learners proofread the most popular subgoal for language and scope.

stage is the most popular subgoal from stage 2, determined by the relative numbers of upvotes and downvotes. Learners are presented with this subgoal label and the steps that it covers, and have the option to refine the label or keep it as is. Ideally, when N learners accept the label without making a change we can assume that the label is final and future learners would be shown this subgoal without being asked a question.

We expect alterations at this stage to be minimal because the necessary changes are often subtle and require a certain understanding of the material. By specifically asking learners to iterate on the subgoal label, we envisioned that some would accept the challenge and improve the quality of the subgoal.

Learner Prompts and the Learning Experience

A major concern in designing Crowdy was to design the workflow such that participation in the process of subgoal label creation would not detract from the learning experience. Our core design decision to use short prompts interspersed throughout the video as the way to collect input from learners was informed by prior research in education and learning technologies, which we review here.

The subgoal learning model suggests that the process of grouping a set of solution steps involves self-explanation [5]. Previous research on self-explanation [2, 10, 26] suggests that when learners are asked to explain examples to themselves, they learn the material better and have greater awareness of their own understanding. Self-explanation encourages learners to engage in deep cognitive processing, which results in more robust learning. Our learner prompts also aim to encourage active learning [25], as opposed to passively watching a video, which is shown to engage learners in the learning process and improve retention of the material.

Research on designing learner prompts shows that learners self-correct misconceptions by answering the prompts [27]. Existing online learning platforms such as Coursera add frequent micro-quizzes inside videos to help people learn better from video lectures, and research on retrieval practice might

provide support for in-video quizzes [12, 13]. In multimedia learning, different learner prompts are shown to provide different pedagogical benefits [21]. The learning gain was higher when learners picked the best explanation than when they were asked to type in a possible explanation [11], possibly because the higher cognitive load in generating an explanation might break the flow in multimedia learning.

These findings suggest that our tasks may actually offer a pedagogical benefit to learners. In future research we intend to evaluate the impact that participation in Crowdy has on learning and — if the impact is positive — to compare it to the benefit conferred by being presented with subgoal labels while watching a how-to video.

PILOT STUDY

To iterate on the workflow design and get a general sense of the quality of learner-generated subgoals, we conducted a pilot study in a user interface design class at a university. The class covers the theory behind designing usable interfaces, and students are responsible for creating and testing a web interface. Therefore, we felt it was a suitable deployment for a series of introductory web programming how-to videos. While students are expected to complete programming assignments and a project involving web interfaces, these topics are not explicitly covered during class time.

There were approximately 280 students in the class, and most were computer science majors. By the end of the class we had brought together 21 videos on HTML, CSS, jQuery/JavaScript, and Meteor (<https://www.meteor.com/>), a JavaScript platform for building web apps) into one place. These videos had an average duration of 6:18 minutes (stdev=2:49). The videos were all existing videos on YouTube. For each video we generated the individual steps and learners went through the three-stage workflow to generate subgoals. There was an average of 76 students participating in each video (stdev=39), 20 subgoals generated per video (stdev=16), and 193 actions (stdev=395), which are logged when learners answer a question or manipulate subgoals in any way. We deployed the videos periodically based on the topics for the upcoming problem sets/projects. The video links were embedded into assignment handouts as a resource. Although students had to be responsible for the material in the videos in order to succeed in the class, watching the videos was optional. We used a group of college students to iterate on our design, but in the future we will consider how using different groups of learners may affect the success of our workflow, especially if there is no constraint on who can participate.

We did not conduct a rigorous evaluation of the results from this deployment, and instead used the periodic deployments as an opportunity to iterate on our design. We found that participation was consistent throughout the semester, although participation varied greatly per video, likely based on video quality and difficulty of the material. In the beginning of the deployment we received a lot of spam answers, which we attribute to unclear instructions on the website that did not emphasize the option to cancel out of a question. Additionally, it was not obvious that a user's answers would be seen by

others. The instructions in our final workflow state that the answers will be seen by other learners, and we added a ‘Cancel’ button to reduce spam.

Design Dimensions Refined through Pilot Study

Through our pilot deployment, we refined important design dimensions for the workflow such as the correct interval to ask questions, whether it is important to show steps, and whether it is important that learners are asked every question. During the deployment, we analyzed the submitted subgoals and talked with users to determine the pain points of our system. For example, we added an interactive tutorial to Crowdy because first-time users commented that they were caught off-guard and confused by the questions. We then iterated on our design and continued our analysis to arrive at our final system.

Question Interval

For the pilot iteration we asked questions at predetermined, fixed intervals because we wanted to focus on the overall design of the questions. We started using a 30-second interval based on our analysis of short (<5 minutes) videos. We arrived at this interval by manually annotating the subgoals for the short videos we used and averaging the time at which the subgoals occurred. However, the web programming videos we used in our pilot deployment were often longer than five minutes. We ended up with a one minute interval for our live deployment, based on users’ feedback that frequent stops can be annoying to them. In the future, we would like to make the question intervals adapt to the domain, context, or video length.

Steps vs. No Steps

We also tested whether the workflow would be successful if the individual steps were not shown, in case step generation becomes a future bottleneck. We found no significant difference between subgoals generated when steps were present and not present. However, users liked the extra information provided and the ability to navigate to different points in the video by clicking on the steps. The presence of steps allowed users who may have lost attention during the video to generate a high-quality subgoal by examining the steps that were covered instead of submitting spam. Therefore we decided to show the steps for the live deployment.

Random vs. Not Random

We also tested whether learners would generate high quality subgoals if they were not asked a question at every interval boundary. Half of the users were asked a question randomly at each interval, and we found that the random interval led to subgoals that varied in scope. One learner who had the random condition and was only asked two questions, both at the very end of the video. Both of the subgoals she generated were summaries of the entire video, instead of being related to the preceding short section. Asking questions at regular intervals tended to lead to greater regularity in subgoals.

LIVE DEPLOYMENT

After improving the workflow based on the pilot study, we publicly launched a website, Crowdy, a wrapper interface to

embed videos and implement our three-stage workflow. The goal of live deployment was to test our claim that learners can collaboratively generate subgoals of similar caliber to those generated by domain experts.

Study Structure

Our study targeted anyone interested in learning introductory web programming. Crowdy was presented as a way to learn web programming through collaborative video watching. In order for the site to stand alone, we included 15 videos (average duration=6:15, stdev=2:08) covering the basics of HTML, CSS, and JavaScript/jQuery. We refined the original selection of videos from the pilot study to remove less popular ones and added some to increase coverage. A summary of the videos included is presented in Table 1. We manually annotated each of the videos to generate a list of individual steps to be shown alongside the video. All of the videos started with zero subgoals, and all contributions were from people visiting the site voluntarily.

Deployment Strategy and Participation

We used various strategies to promote the website. We emailed to lists at universities, posted on social media (Twitter, Facebook, Reddit, Hacker News), reached out to local hacker schools, and bought keywords on Google Ads. Furthermore, we added our link to YouTube comments on the original videos we used to attract learners watching the videos on YouTube to visit Crowdy. In our analysis, we included data from May 5 to May 29, 2014 (25 days).

During the 25-day deployment, 1,268 users visited 2,838 pages, with an average session duration of 1.3 minutes. The top 4 traffic sources were direct access (41.31%, including mailing list clicks and typing in the URL), Twitter (23.81%), Google organic search (12.39%, which are clicks on search results, not advertisements), and Facebook (8.84%). All traffic data was collected with Google Analytics.

Evaluation Method

We chose the four most popular videos on Crowdy (boldface in Table 1) HTML Intro, Intro to CSS, Making divs side by side, and Intro to selectors) to evaluate the quality of learner-sourced subgoal labels. These videos resulted in 17 final subgoal labels.

To judge the quality of the learner-generated subgoals, we compared them to expert-quality subgoals. We recruited two faculty and staff members in computer science at a university, who have extensive experience in web programming, to be our domain experts. After explaining what subgoals are and providing examples, the domain experts watched the four videos that we are evaluating and generated subgoals using our system. Because we only allowed learners to generate subgoals at predetermined intervals, we asked the domain experts to generate labels using the same intervals for direct comparison. In the future, we plan to evaluate subgoals at the entire video level without the set interval. Table 2 compares all the subgoal labels from both the experts and learners. For learner-generated subgoals, we took the subgoals that were in stage 3 or most popular in stage 2 for the four videos we

Video Title	Domain	YouTube Views	Length
Learn HTML- Basic Structure and How to Link Pages	HTML	8,745	4:48
Div and Span	HTML	12,934	6:09
HTML Website Tables & Layouts Tutorial	HTML	210,777	4:19
How to create forms in HTML5	HTML	9,983	9:23
Introduction to styling with CSS	CSS	3,150	6:55
Adding Style Rules Using Google Chrome Developer Tools	CSS	205	4:47
CSS Absolute and Relative Positioning Tutorial	CSS	34,113	6:56
Making Divs Side by Side using CSS	CSS	17,644	4:24
Z-Index CSS Tutorial	CSS	8,158	7:33
Introduction to Selectors	jQuery/JavaScript	40,825	2:06
ID selector	jQuery/JavaScript	36,795	5:53
addClass	jQuery/JavaScript	13,871	4:46
Click Event Handler	jQuery/JavaScript	11,234	6:56
Working on the Drag and Drop Program	jQuery/JavaScript	41,615	9:28
Finishing the Drag and Drop Program	jQuery/JavaScript	38,145	9:16

Table 1. How-to videos used in our live deployment. We evaluated the quality of learnersourced subgoal labels for the four bolded videos because they were the most popular videos on Crowdly.

Subgoals + Steps	Your Evaluation	Subgoals + Steps
Create divs in HTML file	L ✓ R	What HTML code is used to create side-by-side <div>
1. Create a new html file	Please explain:	1. Create a new html file
2. In the body section of the html file, create a div with a id="wrap"		2. In the body section of the html file, create a div with a id="wrap"
3. Create another div inside the wrapper with class="left"		3. Create another div inside the wrapper with class="left"
4. Underneath the left div, create another div with class="right"		4. Underneath the left div, create another div with class="right"

Figure 7. An excerpt from the worksheet given to domain experts to evaluate learner-generated subgoals. The subgoals were randomly placed on the left and right so experts had no notion of which was which. Evaluators were asked to choose whether the subgoals matched, or which was better, and provide an explanation.

selected. At the end of our deployment, five of the subgoals were in stage 3, but most had a clear majority.

To measure how the learner-generated subgoals compare to the expert-generated subgoals, we created a worksheet (Figure 7) to have another group of domain experts rate whether the subgoals matched or which one was better, and why. We randomized which side of the worksheet each of the subgoals was on, so evaluators did not know which were generated by learners and which were by experts. They were instructed to rate based on how well the subgoals covered the given steps, and whether they were concrete and descriptive. We recruited four domain experts who were graduate students in computer science with substantial web programming experience to rate 17 subgoals in the four videos.

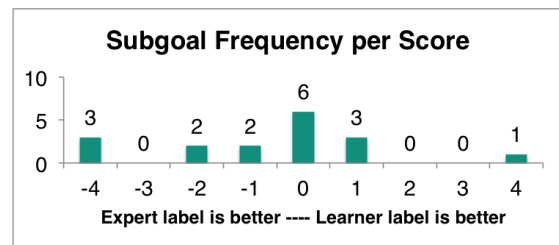


Figure 8. The distribution of subgoal label scores. The scoring scheme assigned -1 if an expert label is better, 1 if a learner label is better, and 0 if they match. While a majority of the subgoals were within the -2 and 2 range, three subgoals had unanimous votes for the expert label, and one subgoal had unanimous votes for the learner label.

We used a scoring scheme to better compare the subgoal labels. Each evaluator's rating is assigned either -1 (expert label is better), 0 (expert and learner label match), or 1 (learner label is better), which match the three options the evaluators were given for each subgoal. The score of -4 means all raters thought the expert label was better, while 4 means all raters thought the learner label was better.

Results

During the 25 day deployment, learners generated 109 subgoals in stage 1, added 14 new subgoals in stage 2, and edited 3 subgoals in stage 3. There were 109 upvotes in stage 2, and 13 upvotes in stage 3. Based on our logging, we found that 119 users (unique sessions) participated in a single video, 17 participated in two, and 14 participated in three or more.

Table 2 shows all 17 labels generated by experts and learners. For 14 out of 17 labels, the learner label was voted as matching or better than the expert label from at least one of the four raters. When we used majority voting to determine a winner in the comparative evaluation, eight labels were considered

Expert Subgoal	Evaluation	Learner Subgoal
<i>Learn HTML- Basic Structure and How to Link Pages</i>		
Create top-level structure of an HTML file	Expert better	The video stopped right as it was starting to talk about the head section
View HTML page in a Web browser	Expert better	Save and open a HTML document in the browser
Add a hyperlink	Learner better	Create link to another page
Create target page of the hyperlink	Match	Create another page to link to and from
View and navigate between both pages	Expert better	Create HTML pages and links
<i>Introduction to styling with CSS</i>		
Understand what CSS is	No majority	Introducing CSS, comparison to HTML
Two kinds of CSS styling	Match	Inline styling, and difference between this and stylesheets
Set text color of a paragraph	Match	Using the style and color attributes of inline styling
Set background color of a paragraph	Expert better	Problems with inline styling, solutions
Create an external stylesheet	Match	Create a stylesheet to set paragraph colors
Linking stylesheet to HTML file	Match	Linking CSS documents to HTML
<i>Making Divs Side by Side using CSS</i>		
Create divs in HTML file	Match	What HTML code is used to create side-by-side <div>
Create CSS for wrapper and left div	Match	Define div appearance with style blocks
Create CSS for right div	Expert better	Style the left and right div, color, height, width
Position divs side by side	Match	Put the two divs side by side and adjust spacing between them
<i>Introduction to Selectors</i>		
Create HTML and JavaScript files	Learner better	Selector introduction
See example of a jQuery ID selector	No majority	Using selectors and jQuery to create event handlers for page objects

Table 2. Expert labels compared to Learner subgoals for the four videos selected for evaluation. For each pair, we identify whether a majority of evaluators felt the subgoals matched or that one was better than the other. In some cases there was no clear majority between evaluators, which we have also noted.

matching between the learner label and the expert label, five were voted in favor of the expert label, two were voted in favor of the learner label, one was a tie between matching and in favor of the expert label, and one was a tie between favoring the expert and learner labels. Figure 8 shows the distribution of scores using the scoring scheme. Due to the highly subjective nature of the matching task, inter-rater reliability (Fleiss' $\kappa = 0.19$) showed a slight level of agreement [18].

We now discuss interesting cases in detail: labels for which the expert label was unanimously voted as better, labels for which the learner label got worse in stage 3, and labels that were voted to be matching.

Example of subgoals where expert's is better

Scoping of a subgoal label matters. Most evaluators judged the subgoals on whether they were accurate representations of the given steps. A common complaint was over subgoals that were too specific, or ones that did not seem to capture the steps at all. Evaluators tended to prefer subgoals that were informative instead of those that captured only generic statements. For example, evaluators preferred the expert subgoal "View and navigate between both pages" over the learner subgoal "Create HTML pages and links" for a subgoal in the introduction to HTML video because the learner subgoal was too broad.

Expert subgoal: View HTML page in a web browser

Learner subgoal: Save and open a HTML document in the browser

Steps:

1. Save file with .html extension
2. Open file in web browser to view

Evaluators felt the expert subgoal was more holistic, whereas the learner subgoal simply enumerated the steps that were covered. This subgoal was added in stage 1, and was chosen over other answers such as "Save and open HTML", "HTML format", "describe page tags", and "Implementation of an HTML page". Out of these choices, "Save and open a HTML document in the browser" is the most accurate and the most specific. However, as a subgoal it is not ideal because it exactly replicates the steps. This subgoal correctly summarizes the steps, as asked in stages 1 and 2, but could have been altered to be less specific in stage 3.

This suggests a lack of instruction on our part in stage 3 where learners are asked to make sure the subgoals cover the provided steps, but are not instructed to make subgoals neither too broad nor too specific. This sort of instruction could be added to stage 3, or we could implement multiple versions of stage 3, where we ask learners about different aspects of the subgoal.

Example where final subgoal got worse in stage 3

Expert subgoal: Create top-level structure of an HTML file

Learner subgoal (most popular subgoal after stage 2): Basic structure of HTML template

Learner subgoal (subgoal altered in stage 3): The video stopped right as it was starting to talk about the head section

In this example, a learner who saw a stage 3 question for this subgoal changed the subgoal to something of much lower quality. The final subgoal in this case does not summarize the video section. Although learners only rarely made a drastic replacement to a subgoal in stage 3, we need to consider mediation methods to prevent this from happening in the future. One possibility is to accumulate votes throughout the stages and only allow minor changes to subgoals that already have a lot of votes.

Example of subgoals that match

Expert subgoal: Set text color of a paragraph

Learner subgoal: Using the style and color attributes of inline styling

Steps:

1. Type 'color' inside the style attribute to set the color of the paragraph
2. Assign the hexadecimal value #FFFF00 to make the paragraph yellow
3. View the file in the browser to see the yellow paragraph
4. Type a semicolon after the property value to separate the properties

These subgoals are slightly different, yet the majority of evaluators felt they accurately covered the steps. This suggests that there can be variability in subgoals. Learners and experts (or people in general) can have different interpretations on what the 'correct' subgoal is. Further study will need to see if learner-generated subgoals, even if imperfect, still lead to improved learning gains as observed in literature.

Interviews with Users

To better understand learners' experiences, we conducted semi-structured interviews with three learners who have used Crowdly. Learners identified themselves to us via email after using our system to ask a question or express their interest in the site. They had various programming experiences, but all were web programming novices. We conducted 30-minute video interviews where we asked them about their experience on the site and tried to learn about their thought process while they answered the questions.

Learners in general felt that being asked questions helped them pay attention, and one specifically noted that answering the questions helped him remember the concepts. This suggests that the learnersourcing prompts may have had a positive effect on learning. Learners also liked being able to reference the video's steps as they watched. According to one learner, *"having the steps there and knowing that I was going to have to fill out goals made me pay attention to the video slightly differently."* In contrast, he felt that if he had watched the video with no steps or questions, he would have mindlessly followed along without learning anything. Another learner commented, *"When I first watched the first*

video I didn't really know what to expect, but in the second and third videos I was more... attentive to watching, to trying to understand what exactly am I watching instead of blindly watching." Learners also recognized that the quality of the underlying video affected how well they learned the material, even if the prompts helped them engage with the material.

Learners had different critical approaches to the multiple choice questions in stage 2. One learner never contributed additional subgoals in stage 2 because he felt *"the ones that were there were pretty descriptive already."* Another added his own answers instead of choosing an existing subgoal about 50 percent of the time. He said, *"I would choose one if it was close to what I would have written... I think I was pretty critical, actually."* Learners liked seeing labels added by other learners. One learner said the choices *"...made me feel as though I was on the same wavelength still."*

The set interval for the learnersourcing prompts was hit or miss. For some videos it seemed okay, but for others it felt too short and others too long. Some learners felt the questions were asked at suboptimal times, especially for the shorter videos. One mentioned *"on a really short video, it was a little harder to see the goals,"* and another said, *"The introduction videos didn't cover very much content, so they could use a longer span, while something that covers more information would need to stop more frequently."*

Interview with Video Creator

We also interviewed a creator of one of the videos to get the author's perspective on the value of having subgoal labels added to the video. We contacted some of the authors through YouTube messaging, and one author volunteered to share his thoughts. We showed him the learner-generated subgoals from his video and asked questions over email. For his video, he felt that most of the subgoals were correct except for the last one, which was too broad in his opinion.

He was happy that the outline and the questions might make learners stay and watch the entire video, or be able to easily find the sections that they want to watch. He commented, *"Having pop up questions means the viewer has to be paying attention."* This means that our system can be beneficial to both video learners and video creators because it has the opportunity to enhance what the author has created with the help of learners watching the video.

DISCUSSION

We discuss the value of having each stage and ways to further improve the workflow, as well as the limitations of our current methodology presented in the paper.

Improving the Workflow for Better Subgoal Labels

Stage 1 is necessary for the system to obtain initial subgoals from scratch. The quality of these initial subgoals largely determines the overall quality of the final subgoal label because future steps mostly focus on refining existing steps, not generating completely new ones. In Figure 9, raw subgoals in stage 1 vary in quality. In our deployment, there were power users who submitted a set of high quality subgoals in stage 1, which usually ended up being the most popular all the

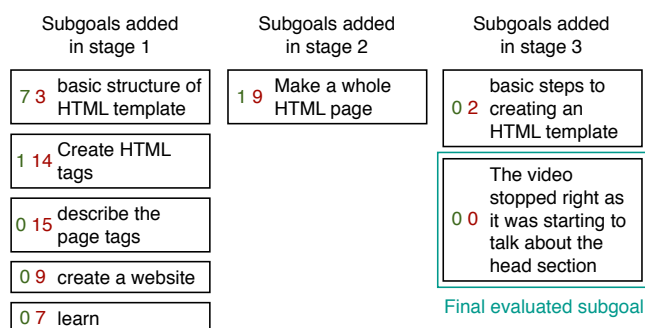


Figure 9. The subgoals that were added in each of the stages for the introduction to HTML video. The green numbers refer to upvotes, and the red numbers refer to downvotes. In stage 3, we expected small iterations, but found that one learner drastically changed the tone of the original subgoal.

way through. How can we better guide learners to come up with quality subgoals? The current learner prompt provides an example, but more contextual, domain-specific examples and counterexamples that highlight the desired properties in a subgoal label might help.

Stage 2 is necessary to remove spam and to pick the best label from candidates, but results suggest that more guidance on which one to pick would improve the voting process. Our current design assumes that one label is going to be significantly better than others, but there is the possibility that they will be close. In such cases, the learner prompt can suggest more specific guidelines by encouraging them to consider scope and language and presenting contrasting good and bad examples. Another potential problem we noticed is that learners are reluctant to submit their own labels in this stage and resort to the best one from the candidates, even when they find the existing ones unsatisfactory. We plan to study the voting process in more depth, and design ways to lower the cost of label revision in stage 2.

Stage 3 is necessary to adjust the scope and language for finalization, but it still triggers learner prompts at a pre-defined subgoal label boundary, preventing learners from looking for video-wide consistency and scoping improvements. Also, in stage 3 we expect learners to only make incremental changes, but a learner who totally missed the point could change a previously ‘good’ subgoal. This is what happened in Figure 9. We are considering mediation methods to prevent later stage input from completely overwriting the earlier stage input and are looking to automatic methods for evaluating subgoals before they are considered final.

In many cases, we found an improvement in subgoals added as learners progressed through the stages. For example, in the ‘Introduction to Selectors’ video, a stage 2 user combined the subgoals ‘create an event handler’ and ‘Introduction to jQuery selectors’ to create the final subgoal ‘Using selectors and jquery to create event handlers for page objects’ (Figure 10). This suggests that even if incomplete, the subgoals in stage 1 are a good baseline for the final selected subgoal.

In general, we found that our learnersourcing workflow succeeded because it allowed many learners to contribute at a

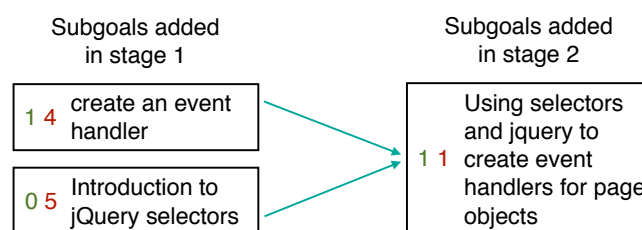


Figure 10. The subgoals that were added in each of the stages for one portion of the introduction to jQuery selectors video. The green numbers refer to upvotes, and the red numbers refer to downvotes. The subgoal added in stage 2 is a combination of the subgoals added in stage 1, which suggests that the learner who created this subgoal may have used the stage 1 subgoals as a baseline.

micro level, and learners were generally successful at identifying the ‘best’ subgoal in each set. For learnersourcing systems in general, we believe it is important to consider the amount of information given to learners at each task so they are equipped to contribute without being distracted from learning. The type of prompt is also important so learners are contributing as much information as possible and still benefiting from the original educational material.

In the future, we would like to make the question interval more context-specific. Some ideas for informed segmentation are to analyze the audio track for speech breaks, run topic modeling on the transcript to detect topic transitions, or leverage signals from previous learners’ in-video interaction [15, 14]. Allowing learners to move subgoals to the correct spot and making sure questions aren’t asked in the middle of words could be a preliminary step. Another approach would be to add additional steps to the learnersourcing workflow to allow learners to contribute the subgoal boundaries.

Only 26% of users participated in multiple videos, so we want to look for ways to encourage learners to stay on the site, perhaps by recommending videos or visualizing the learners’ collective progress.

Limitations

More domains: Our live deployment only looked at web programming videos. The videos we chose were how-to videos that had hands-on, concrete, demonstrated tasks through which instructors walked the learners. Although there is nothing in the workflow that is specific to web programming, we don’t have any quantitative data about learner-generated subgoals from other domains to show that the workflow is domain-independent. We plan to include videos from other how-to domains such as cooking or home improvement.

Learning benefits: While our task design was based on literature in learning sciences, our current evaluation did not directly measure learning improvements from using our workflow. A careful study is needed to explore the tradeoff between learning benefits and distraction or interruption caused by our learnersourcing prompts.

Longer term, larger deployment: Our 25-day deployment with over 1,000 users was helpful for us to get the initial idea of how our workflow works, but we believe a longer and

larger deployment will be helpful. While many people visited the website out of curiosity, not many learners were seriously willing to learn web programming. As a result, not many videos received enough learner input to advance to stage 3. We plan to host more videos and continue the live deployment, which will provide us with more data for a deeper analysis on how learners in the wild use our workflow.

CONCLUSION AND FUTURE WORK

This paper introduces the concept of learnersourcing, a form of crowdsourcing that engages learners in human computation tasks as they interact with educational videos. We built Crowdy, a learnersourcing application for generating subgoal labels from a how-to video, through a multi-stage workflow where learners collaboratively contribute useful information about the video while answering reflective summary questions. Our deployment study shows that a majority of learner-generated subgoals were comparable in quality to expert-generated subgoals, and that learners reported they were paying more attention to the material and felt they understood the material better after going through the workflow.

We envision more advanced learnersourcing applications that enable more constructive and interactive video learning experience at scale. We plan to extend our current workflow to lecture videos on MOOCs that are much less structured than how-to videos. Collaborative summarization on lecture videos might be a useful exercise for learners that deeply engage them in the learning process. Further, we plan to design synchronous learnersourcing workflows for more direct interaction between learners, which can result in more complex and creative outcomes.

ACKNOWLEDGMENTS

The authors thank Philip Guo, Max Goldman, Lauren Margulieux, Mark Guzdial, and the UID Group at MIT CSAIL for suggestions and feedback. This work was supported in part by the MIT EECS SuperUROP Program and Cisco Systems, Inc., by Quanta Computer through the Qmulus project, by NSF award SOCS-1111124, and by a Sloan Research Fellowship. Juho Kim is partly supported by the Samsung Fellowship.

REFERENCES

1. 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 *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, ACM (New York, NY, USA, 2010), 313–322.
2. Bielaczyc, K., Pirolli, P. L., and Brown, A. L. Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem solving. *Cognition and instruction* 13, 2 (1995), 221–252.
3. Buykx, L., and Petrie, H. Recipe sub-goals and graphs: An evaluation by cooks. In *Proceedings of the ACM Multimedia 2012 Workshop on Multimedia for Cooking and Eating Activities*, CEA '12, ACM (New York, NY, USA, 2012), 59–64.
4. Catrambone, R. The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General* 127, 4 (1998), 355.
5. Catrambone, R. Task analysis by problem solving (taps): uncovering expert knowledge to develop highquality instructional materials and training. In *2011 Learning and Technology Symposium* (2011).
6. Chi, P.-Y., Ahn, S., Ren, A., Dontcheva, M., Li, W., and Hartmann, B. Mixt: Automatic generation of step-by-step mixed media tutorials. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, ACM (New York, NY, USA, 2012), 93–102.
7. Cross, A., Bayyapunedu, M., Ravindran, D., Cutrell, E., and Thies, W. Vidwiki: Enabling the crowd to improve the legibility of online educational videos. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '14, ACM (New York, NY, USA, 2014), 1167–1175.
8. Eiriksdottir, E., and Catrambone, R. Procedural instructions, principles, and examples how to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 53, 6 (2011), 749–770.
9. Grossman, T., Matejka, J., and Fitzmaurice, G. Chronicle: Capture, exploration, and playback of document workflow histories. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, ACM (New York, NY, USA, 2010), 143–152.
10. Hausmann, R. G., Nokes, T. J., VanLehn, K., and Gershman, S. The design of self-explanation prompts: The fit hypothesis. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society* (2009), 2626–2631.
11. Johnson, C. I., and Mayer, R. E. Applying the self-explanation principle to multimedia learning in a computer-based game-like environment. *Comput. Hum. Behav.* 26, 6 (Nov. 2010), 1246–1252.
12. Karpicke, J. D., and Blunt, J. R. Retrieval practice produces more learning than elaborative studying with concept mapping. *Science* 331, 6018 (2011), 772–775.
13. Karpicke, J. D., and Roediger, H. L. The critical importance of retrieval for learning. *science* 319, 5865 (2008), 966–968.
14. Kim, J., Guo, P. J., Cai, C. J., Li, S.-W. D., Gajos, K. Z., and Miller, R. C. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, ACM (New York, NY, USA, 2014), 563–572.

15. Kim, J., Guo, P. J., Seaton, D. T., Mitros, P., Gajos, K. Z., and Miller, R. C. Understanding in-video dropouts and interaction peaks in online lecture videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference, L@S '14*, ACM (New York, NY, USA, 2014), 31–40.
16. Kim, J., Nguyen, P. T., Weir, S., Guo, P. J., Miller, R. C., and Gajos, K. Z. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems, CHI '14*, ACM (New York, NY, USA, 2014), 4017–4026.
17. Kong, N., Grossman, T., Hartmann, B., Agrawala, M., and Fitzmaurice, G. Delta: A tool for representing and comparing workflows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, ACM (New York, NY, USA, 2012), 1027–1036.
18. Landis, J. R., Koch, G. G., et al. The measurement of observer agreement for categorical data. *biometrics* 33, 1 (1977), 159–174.
19. Lasecki, W. S., Song, Y. C., Kautz, H., and Bigham, J. P. Real-time crowd labeling for deployable activity recognition. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work, CSCW '13*, ACM (New York, NY, USA, 2013), 1203–1212.
20. Margulieux, L. E., Guzdial, M., and Catrambone, R. Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research, ICER '12*, ACM (New York, NY, USA, 2012), 71–78.
21. Mayer, R. E., and Moreno, R. Nine ways to reduce cognitive load in multimedia learning. *Educational psychologist* 38, 1 (2003), 43–52.
22. Noronha, J., Hysen, E., Zhang, H., and Gajos, K. Z. Platemate: Crowdsourcing nutritional analysis from food photographs. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST '11*, ACM (New York, NY, USA, 2011), 1–12.
23. Pavel, A., Berthouzoz, F., Hartmann, B., and Agrawala, M. Browsing and analyzing the command-level structure of large collections of image manipulation tutorials. Tech. Rep. UCB/EECS-2013-167, EECS Department, University of California, Berkeley, Oct 2013.
24. Pongnumkul, S., Dontcheva, M., Li, W., Wang, J., Bourdev, L., Avidan, S., and Cohen, M. F. Pause-and-play: Automatically linking screencast video tutorials with applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST '11*, ACM (New York, NY, USA, 2011), 135–144.
25. Prince, M. Does active learning work? a review of the research. *Journal of engineering education* 93, 3 (2004), 223–231.
26. VanLehn, K., Jones, R. M., and Chi, M. T. A model of the self-explanation effect. *The journal of the learning sciences* 2, 1 (1992), 1–59.
27. Williams, J. J., Kovacs, G., Walker, C., Maldonado, S., and Lombrozo, T. Learning online via prompts to explain. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems, CHI EA '14*, ACM (New York, NY, USA, 2014), 2269–2274.