When Crowds Come Together: Supporting Engagement and Peer Learning in a Classroom Setting

Benjamin B. Bederson
Human-Computer Interaction Lab
Computer Science Dept.
University of Maryland
www.cs.umd.edu/~bederson

Anne Rose
Human-Computer Interaction Lab
University of Maryland
rose@cs.umd.edu

ABSTRACT
Massive Open Online Courses (MOOCs) have been very effective at bringing attention to technology and learning. But, their focus on remote, asynchronous situations leaves a gap for the co-present, synchronous settings of most university classrooms. This paper investigates the use of technology IN classrooms to better support active student engagement. By harnessing student effort with a human computation model, we provide a tool called XParty that supports a pedagogically useful activity that simultaneously engages the entire class and gives students and the instructor alike feedback about what students are thinking.

Author Keywords
Education; massively open online courses; collaborative learning; learning; human computation; crowdsourcing.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
One of the nice side effects of the recent interest in MOOCs (Massive Open Online Courses) is an increase in thinking about active learning in the classroom, and how to decrease traditional lecture. One challenge that remains is how to scale up student interaction. The easy approach of having faculty ask questions of their students has a number of problems including the fact that it isn't scalable. Furthermore, students that do answer tend to be the self-selected assertive ones. Individual students often go through an entire semester without ever raising their hands. A more scalable approach is through peer learning. This has small groups of students work together – who then typically report back en masse through "clickers" or a website. This is scalable, but the level of student expression is typically limited to various kinds of multiple choice. This may be minimally sufficient for learning technical subjects, but doesn't support creative and generative thinking well.

Instead, we are looking at a model that is both scalable and supports richer creative expression. In this model, students (individually or in groups) report their thinking via a computer by writing prose or (conceptually) uploading an image of a visual design they may have created on paper. Then, a hybrid human-computer solution aggregates and clusters the student input so the instructor and the students together can see and make sense of what the students are thinking.

We are currently building this hybrid solution that uses a computer-based task distribution and aggregation process with student ("crowd") support where the students themselves provide the individual work. Since looking at and analyzing other student's efforts is a pedagogical activity in itself, the result is that this approach results both in students learning about what other individual students are doing as well as resulting in a "report" that describing what the class as a whole is collectively thinking about.

MOTIVATION
Recent activity around MOOCs [8] allow teachers to deliver lectures to students outside of the classroom, allowing them to spend more time in classes actively engaging in problem solving, discussions, or collaborating on hands-on activities. This so-called "flipped classroom" structures events that traditionally take place in the classroom outside the classroom, allowing more time for other in-class activities [5]. Because teachers are not lecturing during class time, a higher degree of interactivity between students and teachers is possible.

We are further motivated by learning theory such as "Situated Cognition" by Brown, Collins and Duguid [2] who argue for the importance of authentic learning experiences by designing learning activities that directly connect (or "situate") the learning activity to how the thing being learned would be used in a more natural setting. They further argue for the importance of group learning structures: "Groups are not just a convenient way to accumulate the individual knowledge of their members. They give rise synergistically to insights and solutions that would not come about without them." [2, p. 40].

We found the work of Morris and Horvitz on SearchTogether [6] and Moraveji et al. on ClassSearch [7] compelling. SearchTogether allows groups of people, geographically remote from each other, to collaboratively search, synchronously or asynchronously. While not used in classroom environments or designed to support flipped classrooms, SearchTogether provided inspiration for how

people could synchronously work together in a modern web-based environment to elaborate on each other's work.

ClassSearch also supports collaborative search activity, but its focus was on deployments in classroom environments including support for aggregating student activity by showing a dashboard that students and instructors together could view on a shared display to learn from each other.

We initially recreated much of the native ClassSearch app as an entirely web-based tool we called SearchParty [1], and confirmed the potential of this approach [4]. The code for SearchParty is completely open source, and is publicly available at https://search-party.googlecode.com. In addition, the service is freely available for all to use at http://search-party.appspot.com.

It was only after our experience exploring the support of students learning to search that we broadened our efforts to support learning in a broader set of disciplines. It was this broadening of SearchParty that led to the name of our current approach: XParty.

OUR DESIGN

Just as SearchParty let students do search tasks and aggregated search results, XParty lets students do the broader task of responding to any instructor question and aggregates the student responses. Unlike the structured activity of web search or multiple choice, XParty (Figure 1) provides a text box for students to enter prose (and conceptually would also support uploading images.) The challenge is how to aggregate arbitrary textual input in a meaningful way that can be used rapidly in the context of a fast-paced dynamic classroom.

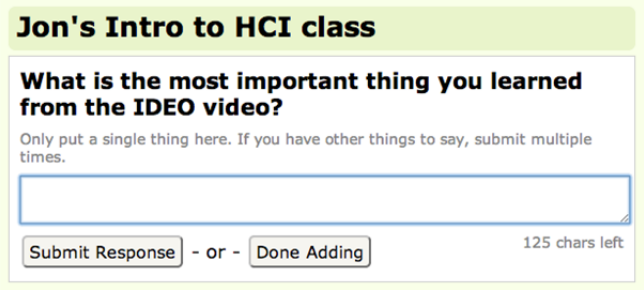

Figure 1. Initial student task in XParty. Students are asked to answer an instructor's question on a simple website through a standard textbox on a web form.

Early Design Approaches

Given our focus on supporting peer learning, our designs of SearchParty as well as initial designs of XParty had a real-time display of student activity that students could see on their own computers, and that was intended for display on the shared classroom display. This was fun and definitely engaging – but we consistently found in classroom trials that when students were starting the activity and should focus on their own thoughts initially, they were consistently distracted by the shared view, and the novelty of seeing what other students were doing – thus making it harder for them to develop their own independent thoughts. Thus, in the later versions, we removed the ability to see what everyone was doing during the initial activity.

Early Categorization Approaches

To categorize student responses, we initially tried several techniques that would automatically create the best quality categorization for immediate use in the classroom. We tried the following approaches:

- **Automated categorization:** While fast, the computational linguistic algorithms we explored did not create useful categories from the student responses because of several problems. The first was that the clusters of student responses were not labeled. Having clear labels is crucial since a teacher must be able to quickly look at a group of responses and know why they go together. We tried asking students to create the labels but this proved difficult because of the wide variety of responses included within a cluster. The second problem was that most clustering algorithms were designed for larger documents, being based on "bags of words" techniques. Finally, most algorithms we found required the specification up front of the number of categories or clusters, but there was no rationale way to make this decision.
- **Tagging responses:** We asked students to create tags (or categories) for individual responses, and then tried to cluster the responses based on these tags. Again, the clusters created were not very useful – due both to the problems mentioned with the previous approach as well as the fact that students gave a wide variety of tags to each student contribution.

While our initial approaches were fast, they did not create usable categories so we decided to adapt Cascade [4], a recent human computation based solution categorizing data.

Cascade

Cascade is a crowd-sourced approach for creating high quality taxonomies of existing information. Workers perform three basic types of tasks: 1) generating categories; 2) selecting the best categories; and 3) voting on whether or not items fit the categories. Cascade is designed around the assumption that workers are somewhat unreliable, and while aiming for a reasonably efficient solution, optimizes for quality. It uses task redundancy to get input from multiple workers, and voting thresholds to decide if the requirements of a certain task have been met. If the resulting taxonomy is not sufficient (i.e., there are too many uncategorized items or some items only exist in large categories), the process is run again on these items.

Cascade uses 5-fold redundancy by default and was designed for workers on Amazon Mechanical Turk. The resulting taxonomies are of high quality but can take several hours to complete – in large part due to workers not performing tasks perfectly in parallel.

To be usable in a classroom setting, we needed to adapt Cascade to the classroom context which resulted in the following conceptual changes:

- **Students are more reliable:** In the context of a face-to-face classroom where individual students are identified, we have found that in practice most students put in an honest and strong effort.
- **Students can't wait:** Given that this is fundamentally a learning activity, it is unacceptable for students to have to wait between phases while other students finish their assigned tasks.
- **Tasks much be pedagogically meaningful:** In order for students and the instructor alike to feel like it is worth a few minutes to do the categorization, the tasks can not feel too boring, redundant or meaningless.

Current Implementation

XParty allows students to submit multiple answers to a question and then indicate when they are done. As soon as a student is done submitting answers, they start receiving one of the three types of categorization tasks until the taxonomy can be created.

Since the students created the items being categorized and thus have good domain knowledge, and are motivated by the social setting of the classroom and teacher, we reduced the time required to run Cascade by decreasing the task redundancy and removing the iteration. Also, since there are roughly 7 times more category "fit" tasks than the other two types of tasks, we chose to use a smaller redundancy for category fit tasks. The task redundancy used by XParty is automatically calculated based on the number of students. Cascade uses a two-step category fit task where the second step verifies the categories that were marked as fitting in first phase. We removed the verification step in XParty.

When we initially adapted Cascade for use in XParty, students performed tasks in parallel but they needed to wait until one task type was finished before performing the next type of task. This was especially problematic if students were working at different speeds since faster users might have to wait significant periods before continuing their work. And one slow student using a mobile phone could make the entire class wait for him/her. To reduce student waiting, we modified XParty so students could work on any of the three types of tasks at any time – with the recogition that the quality of the resulting categorization might be somewhat reduced since some category analysis would be done before all the generated categories were available.

While we have attempted to reduce the time required as much as possible, an instructor may have limited time to devote to this exercise in class. XParty allows an instructor to generate the taxonomy before all the categorization tasks are complete. The instructor dashboard (Figure 2) shows how many answers have been submitted and the percentage of categorization tasks that have been completed so an instructor can decide if/when they want to force the taxonomy to be created.

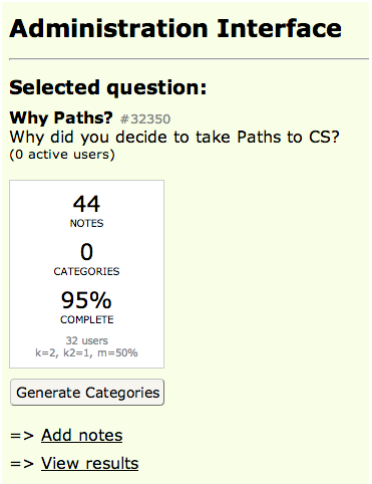

Figure 2. Instructor dashboard shows how categorization tasks are proceeding in case the instructor jump to see the generated categories.

Users can expand and collapse the resulting categories and show or hide nested subcategories. Similar categories and nested subcategories are automatically detected based on the size ratio of the two categories and the percentage of overlapping responses. Specifically, two categories are merged if one is no more than twice the size of the other, and at least 80% of the smaller one's items are in the larger one. A category is considered a subcategory of another if it is at less than half the size of the other, and at least 80% of its items are in the larger group.

A response can appear in multiple categories. While this can be useful, it can be overwhelming for instructors trying to quickly understand the results especially when there are large categories that contain high percentages of the overall responses. To help address this issue, we added an option on the results page that allows instructors to only display each response in a single category. The category chosen is the smallest category or subcategory that the response appears in with preference given to subcategories.

XParty is written as a Python web app hosted by Google App Engine. It is available for download as open source code at https://github.com/bederson/qa/wiki.

RESULTS

We tested XParty in a half dozen trials in various settings of 10-40 people. This includes audiences in 3 talks, 2 pilot tests in lab settings, and 1 use in the first author's class of 40 students. The software was refined after each trial, with the description in this paper being used only in the final classroom trial.

In this trial, the 40 undergraduate students were asked why they chose to take this particular introduction to computer science class? The entire exercise including describing the activity and explaining the website, answering the question,

doing the categorization and discussing the results took just under 10 minutes. The students generated 44 answers which they put into 14 categories. The generated categories (with number of categorized items in parentheses) were:

- Language (7)
- Grades (7)
- Major (6)
- Scheduling (6)
- Leisure (5)
- Flexible (2)
- New programming language (1)
- Professional Development (1)

As an example, answers that were categorized as "Scheduling" were:

- "Fit my schedule and I wanted to try out CS"
- "I had no previous experience with CS and I wanted to get to know the field."
- "I LOVE the self-directed aspect of the class."
- "I thought it would be more hands on."
- "less time pressure"
- "Met at a good time for the rest of my schedule as compared to 122 or 131"

CONCLUSION

While this work remains ongoing, we have developed enough experience with the approach that we are enthusiastic about its potential. To summarize, our approach towards supporting classroom engagement is:

- All students in the class participate.
- Students produce creative content (not just selections from multiple pre-defined options).
- Students see and analyze each other's work, resulting in an aggregation that can result in a valuable understanding of many viewpoints.

The basic requirements for this approach to be successful (which the current solution meets) are:

- Minimal setup and instruction to students.
- Short entire activity.
- Student analysis tasks are themselves meaningful, so time spent is worthwhile.
- Solution must be general enough to support a wide range of disciplines and activities.
- The outcome must be a simple and readily understandable aggregate view of the students' work.

Going forward, our short term plan is to role out the tool more broadly so we can study it in a wider range of contexts. As we fine tune its efficacy, we can start to study specific characteristics of the tool such as the impact of whether student contributions are anonymous or pseudonymous (students are currently identified by their Google account.) And finally, we would like to eventually consider whether this approach could be scaled up to much large student groups such as might be found in asynchronous MOOC settings, or generalized to other settings such as brainstorming among small, synchronous, face-to-face groups which has many of the same requirements as the problem we address here.

ACKNOWLEDGMENTS

Thanks to Google and Huawei for support of this project.

REFERENCES

1. Bederson, B. B., Quinn, A. J., & Rose, A. (2012). SearchParty: Learning to Search in a Web-based Classroom, in Proceedings Of *ACM Conference on Human Factors in Computing Systems (CHI 2012)*, Workshop paper. https://sites.google.com/site/eist2012/
2. Brown, J. S., Duguid, P. 1989. Situated cognition and the culture of learning. *Educ. Researcher*, 18(1), 32-42.
3. Chilton, L. B., Little, G., Edge, D., Weld, D. S., and Landay, J. A.. 2013. Cascade: crowdsourcing taxonomy creation. In Proceedings of the *SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1999-2008.
4. Gubbels, M., Rose, A., Russell, D., Bederson, B. B. (September 2012). SearchParty: Real-time Support for Social Learning in Synchronous Environments. Tech Report #HCIL-2012-21, University of Maryland Human-Computer Interaction Lab. http://www.cs.umd.edu/localphp/hcil/tech-reports-search.php?number=2012-21
5. Lage, M. J., Platt, G. J., Treglia, M. Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment. *Journal of Economic Education*, 31(1):30–43, Winter 2000.
6. Morris, M. and Horvitz, E. (2007) SearchTogether: an interface for collaborative web search, in Proc. of *User Interface Science and Technology (UIST 2007)* , ACM Press, New York, pp. 3–12.
7. Moraveji, N., Morris, M., Morris, D., Czerwinski, M., and Henry Riche, N. (2011) ClassSearch: facilitating the development of web search skills through social learning in Proc. of *Conference on Human Factors in Computing Systems (CHI 2011)*, ACM Press, New York, pp. 1797-1806.
8. Russell, D. M., Klemmer, S., Fox, A., Latulipe, C., Duneier, M., and Losh, E.. 2013. Will massive online open courses (MOOCs) change education?. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. ACM, New York, 2395-2398.
9. Shneiderman, B., Borkowski, E., Alavi, M., Norman, K. (1998) Emergent Patterns of Teaching/Learning in Electronic Classrooms Educational Technology *Research and Development* 46, 4, pp. 23-42.