Crossroads The ACM Magazine for Students

WINTER 2010 VOL.17 · NO.2

XRDS

Comp-YOU-ter

Challenges and Opportunities in Human Computation

Crowdsourcing as Democratization of Data Collection

Profile: ReCaptcha's Luis von Ahn

XRDS.ACM.ORG

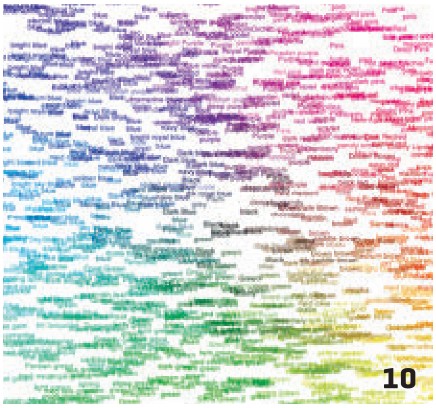
Association for Computing Machinery

XRDS

Crossroads
The ACM Magazine for Students
WINTER 2010 VOL.17 · NO.2

Comp-YOU-ter

5



10

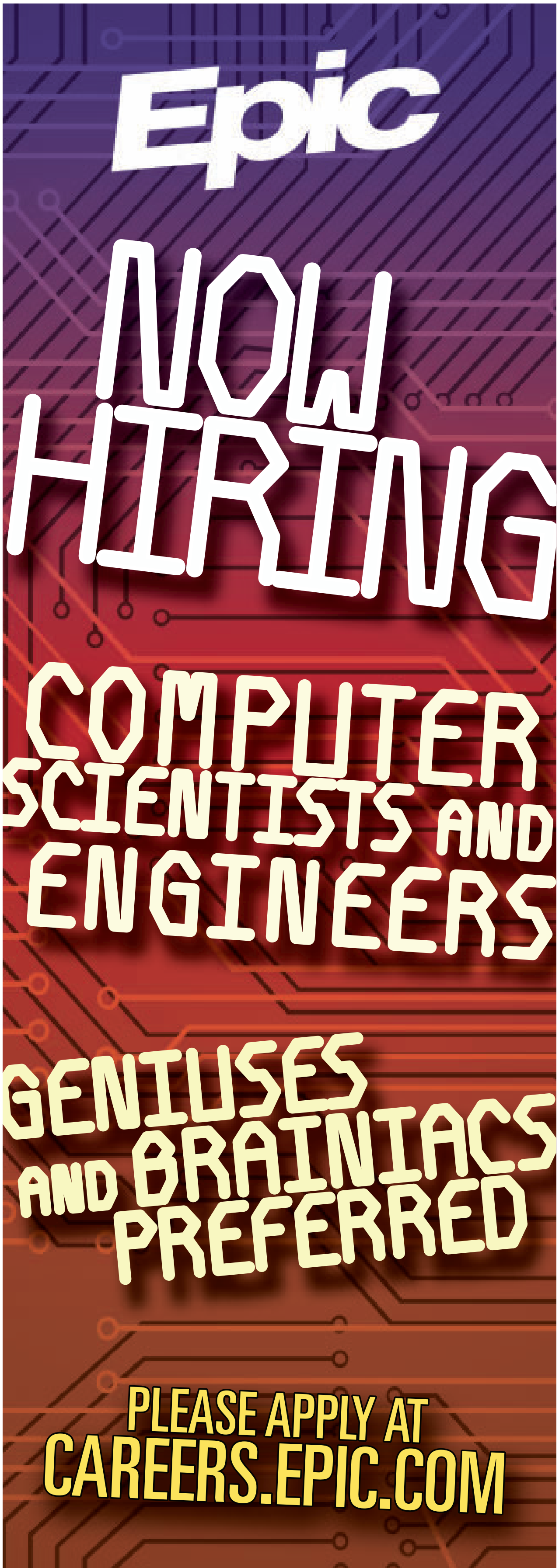
55

begin

features

end

Association for Computing Machinery



XRDS

ACM
Association for Computing Machinery
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA
+1 212-869-7440

CONTACT
General feedback: xrds@acm.org

For submission guidelines, please see http://xrds.acm.org/authorguidelines.cfm

SUBSCRIBE

Subscriptions ($19 per year) are available by becoming an ACM Student Member www.acm.org/membership/student

Non-member subscriptions: $70 per year http://store.acm.org/acmstore

ACM Member Services
To renew your ACM membership or *XRDS* subscription, please send a letter with your name, address, member number and payment to:

ACM General Post Office
P.O. Box 30777
New York, NY
10087-0777 USA

Postal Information
XRDS: Crossroads (ISSN# 1528-4972) is published quarterly in spring, winter, summer and fall by Association for Computing Machinery, 2 Penn Plaza, Suite 701, New York, NY 10121. Application to mail at Periodical Postage rates is pending at New York, NY and additional mailing offices.

POSTMASTER: Send addresses change to: *XRDS: Crossroads*, Association for Computing Machinery, 2 Penn Plaza, Suite 701, New York, NY 10121.

Offering# XRDS0172
ISSN# 1528-4972 (print)
ISSN# 1528-4980 (electronic)

LETTER FROM THE EDITOR

Names on the Page
XRDS, Networking and You

If you've subscribed to this magazine for a while, you've probably noticed *XRDS* has shifted to highly thematic issues. The reason for this is two-fold: we hope by the time you put down the magazine, you'll be able to 1) speak intelligently about the field and 2) know the movers and shakers.

Although you probably speed through the articles not paying much attention to the authors, they're worth a second look. You'd be surprised how often you bump into these people in airport terminals, conference hotels, or wandering the halls of your university following a guest lecture or faculty summit. If you strike up a conversation by asking, "So, what do you work on?" it can be a real killer and missed opportunity. That's like bumping into Steve Jobs and asking, "So, are you, like, into technology?"

Fumbled Opportunities

I remember stumbling into a dedication event during my second week at Carnegie Mellon University and casually being introduced to a professor (whom I had never heard of) named Randy Pausch. I shook his hand, said hello, and meandered over to the free food, not thinking much of it. Then people started filing out to go to some "last lecture" he was giving. I had some free time and figured I'd tag along with the crowd.

A few minutes later, and whoa.

If you don't yet know the name Randy Pausch or what his Last Lecture was all about, it really is worth an hour of your time. The lecture is very easy to find on YouTube.

This sort of bumping into important people without knowing at the time why they are significant to me and my own work has happened a few too many times for comfort. Another serendipitous encounter happened when I was giving a presentation on tricky progress bar manipulations, only to discover the person who wrote the seminal paper on progress bars, Brad Myers, was sitting in the audience.

"Maintaining a high quality bar of authors—people you would actually want to meet—is one of our primary missions in serving you, our readers."

Another time, in search of some travel advice before departing for Guatemala, I ended up conversing with Luis von Ahn, who hails from the country. Good thing I knew who he was before totally embarrassing myself! I was subsequently able to twist his arm for an interview in this issue; see page 49.

When we brainstorm themes for future issues, we spend a lot of time thinking about not only what topics are interesting, but also who the interesting people are in that field. That's whom we contact first and ultimately end up as the authors of the articles you read in *XRDS*. These are also the people you should contact for research projects, internships, post docs, and so on. We've been very fortunate to have top researchers contribute to every issue. Maintaining a high quality bar of authors—people you would actually want to meet—is one of our primary missions in serving you, our readers.

XRDS is Thinking Green

Speaking of brainstorming themes, we've nailed down the next two issues. The first is a comprehensive look at where banking, currency, finance, trade, and related items are heading. Money and the exchange of money is one of the oldest technologies we continue to use today—more than 5,000 years old—and practically on a daily basis. Computing technology has already clearly revolutionized how we exchange money, but we're set for another leap forward.

Following that, we're tackling green (environmental) technologies: intelligent power grids, electric vehicles, innovative recycling processes, robotic street cleaners… All these technologies rely on scores of computer scientists to become reality. So although it's a little off the beaten CS track, it's really relevant, and it's a rapidly growing sector (if you're thinking about jobs). It's also a feel-good industry, where you can innovate, improve people's lives, and help mother earth.

If you've got ideas for articles or top-notch contacts you want us to connect with, shoot us an email at xrds@acm.org. This really is your magazine, and we like to follow a *crowd-sourced approach*, which you'll be reading all about in this issue.

Chris Harrison

Chris Harrison, editor-in-chief is a PhD student in the Human-Computer Interaction Institute at Carnegie Mellon University. He completed his BA and MS in computer science at New York University in 2005. Harrison is a recipient of a Microsoft Research PhD Fellowship. Before coming to CMU, Harrison worked at IBM Research and AT&T Labs. He has since worked at Microsoft Research and Disney Imagineering. Currently, he is investigating how to make small devices "big" through novel sensing technologies and interaction techniques.

INBOX

In General...

XRDS is fantastic. [To Chris Harrison,] Congrats on the editor-in-chief position. Keep that intellectual nutrition coming. I am famished!
Tanimu DeLeon-Nwaha, *Email*

Here is my feedback regarding the latest issue of *XRDS*: please don't use the "continued on/from page..." structure. It's really confusing. I am sure you had some real reasons for that, but it's making the reading less friendly. Even less friendly when the ends of two different articles are on the same page (like page 50 of vol. 17, no. 1, for instance). Other than that, I enjoyed reading the magazine. Thank you for your work.
Dmitry Knyaginin, *Email*

ACM Benefits

Whoa, print version of the @XRDS_ACM magazine? I didn't know I had paid for that. Thank you, @TheOfficialACM!
htowninsomniac, *Twitter*

Future Issues

Why don't you consider [dedicating an issue of *XRDS* to] e-learning and perhaps m-learning? Both sound to be cutting-edge research areas in computer science. Also you may include something like ubiquitous computing... Is it too vast? Then you might want to consider automatic essay grading and plagiarism detection in e-learning...
Arun Sambhanthan, *Facebook*

What about an issue on mobile and ubiquitous computing where you can actually explore how mobiles are being used for social change? Also, there is quite a good opportunity for ICTD (information and communication technology for development). People are developing some low-cost technologies, text-free user interfaces, community radio to extend technology to the masses. In the last four years, a new conference ICTD (www.ictd2010.org) has been created, which is supported by both ACM and IEEE.
Kuldeep Yadav, *Facebook*

Join this conversation. Post your ideas for future issues to our Facebook Group

Whoa, print version of the @XRDS_ACM magazine? I didn't know I had paid for that. Thank you, @TheOfficialACM!

htowninsomniac, *Twitter*

Discussion page: http:// tinyURL.com/XRDS-Facebook

A Word About Email

When sending the *XRDS* e-newsletters, one point to consider: Google [Gmail] stacks messages with identical subject lines and from same senders into "conversations," which I've found to be one of Gmail's weaker attributes. It's easy to lose sight of a new message (sometimes the recipient assumes it's just an old message because the conversation had already reached a natural conclusion. I've even seen cases myself where the new message does not show up at the top of my inbox and is relegated to the date position of the prior message). Since many of your members have Gmail accounts, you might consider using a new, unique (issue-numbered?) subject line with each edition you send out.
Dave Paulson, *Email*

How to contact *XRDS*: Send a letter to the editors, an amazing photo that highlights your work or something interesting on your campus, or other feedback by email (xrds@acm.org), on Facebook by posting to our group page (http://tinyurl.com/XRDS-Facebook), via Twitter by using #xrds or @XRD_ACM in any message, or by post to ACM Attn: XRDS, 2 Penn Plaza, Suite 701, New York, NY 10121, U.S.

PHOTO SPOTLIGHT

Thousands of Chinese workers assemble and test fiber optic systems. In many places of the Chinese economy, human labor replaces automation.

Photo by Jurvetson (Flickr).

begin

INIT

Human Computation and Crowdsourcing

In 1937, Alan Turing formalized the notion of computation by introducing the Turing machine, thus laying the theoretical foundations of modern computer science. Turing also introduced a stronger computational model: a Turing machine with an oracle.

In addition to performing computations itself, such a machine is able to ask the oracle questions and immediately receive correct answers, even if these questions are too hard for the machine itself to compute. Depending on the oracle's capabilities, a Turing machine with an oracle therefore could be much stronger than the machine on its own. The oracle itself is an unspecified entity, "apart from saying that it cannot be a machine" (from Turing's 1939 work, *Undecidable*).

The concept of a Turing machine with an oracle is purely mathematical, yet it springs to mind when observing how today's computers use human capabilities in order to solve problems. Computers are now able to complete tasks that involve challenges far beyond what algorithms and artificial intelligence can achieve today—recognizing anomalies in photos, solving a Captcha puzzle, or judging artistic value—by outsourcing these challenges to humans. Millions of people being on the internet is what makes this outsourcing possible on a large scale. It's known as crowdsourcing.

Understanding these reversed forms of human-computer symbiosis, in which the computer asks a person to compute something instead of vice versa, is the main object of research in the rising field of human computation.

SYMBIOSIS

While we say that computers are using human capabilities, of course it's really humans who are using computers in order to utilize other humans' capabilities. Indeed, in many applications the role of computers is simply to coordinate between humans as they interact among themselves.

This aspect of human computation can be described as a novel form of social organization, in which computers are mediators. The most prominent example is Wikipedia, where computers serve as the platform for aggregation of knowledge and efforts by many people, making it possible to produce a vast, comprehensive, and coherent encyclopedia equivalent to a printed book of around 1,000 volumes, all in a completely distributed manner!

The articles in this issue of *XRDS* provide striking answers to major questions in the field of human computation and crowdsourcing. At the same time, they generate even more questions, highlighting future research directions and opportunities to get involved in the field.

INCENTIVES

Do people partake in human computation tasks for fun, like playing a game? Luis von Ahn of Carnegie Mellon University thought so when he developed Games with a Purpose (see page 49).

Are people motivated by money? Sharon Chiarella, vice president at Amazon, shares her view on the human computation market she leads, Mechanical Turk (page 50), a controversial site where people can pick up micro-tasks and get paid to do them.

Panagiotis Ipeirotis of New York University has been collecting data about Amazon's online marketplace and presents on page 16 a fascinating new analysis of the system.

Or perhaps there's no need for any special incentive. David Ayman Shamma of Yahoo! Research shows how human computation can happen passively, taking advantage of everyday human activities like using Twitter in order to complete computational tasks such as video parsing (page 36).

CURRENT AND FUTURE APPLICATIONS

Robert Miller of MIT and his colleagues outline the major challenges involved in building systems that use human intelligence as an integral component, including understanding what kinds of problems are appropriate for human computation, and dealing with specific software-design and performance challenges arising when humans are in the loop (page 27).

SOCIAL AND ETHICAL IMPLICATIONS

Six Silberman, Lilly Irani, and Joel Ross, graduate students at University of California-Irvine, view Mechanical Turk through the eyes of the workers. They describe some of the problems the voluntary workforce experiences and present several approaches to increase fairness and equality (page 39).

We hope you enjoy this issue of *XRDS*! Let us know by emailing xrds@acm.org.

—Michael Bernstein and Inbal Talgam-Cohen, Issue Editors

5,400

Number of people the winning team (MIT) reached in 36 hours using a recursive incentive recruiting algorithm during the DARPA Red Balloon challenge.

$200 USD

The cost to have 10,000 different left-facing sheep drawn for you using Amazon's Mechanical Turk.*

*http://www.thesheepmarket.com

BENEFIT

ACM Career and Job Center

The ACM Career and Job Center is a web site set up just for ACM members. Getting a job is something all final-year students will be deeply concerned about. The benefit of the ACM job board is it's concerned with only one field: computer science.

So what does the site offer? It's split into two main areas: Job Seekers and Employers. Within the job seekers section, there are three main areas of concern to students: searching for jobs, posting a resume, and job resources.

The job search is fairly standard, with the ability to search by keyword, location, or job type. At the time of this writing, there were 125 jobs listed. As a warning to non-U.S. students, 108 of these jobs were in the United States. They cover a broad range of careers, with a slight focus on academic jobs. You can set up automated alerts so that you don't have to visit the site every day to check for new jobs.

The Resources section is the weakest of the three, not only being extremely U.S.-centric but also somewhat disorganized. Although there is a substantial amount of information, you have to dig for it.

The final section offers the ability to post a resume for potential employers to view.

While I don't advocate using the ACM Job Center exclusively, it does provide another place to look: http://jobs.acm.org.

—*Daniel Gooch*

ADVICE

The Academic Job Search

How to Prepare Key Documents

The academic job search lasts about nine months—after years of preparation! It's a grueling process, and the chances for failure are high.

Before setting out on an academic job search, it's crucial that you prepare a number of key documents that will help you sell yourself as a candidate and future faculty member. This checklist and related advice can help you prepare the right application materials.

1. Cover letter, CV, research statement, teaching statement. I call these the primary documents. The book *Tomorrow's Professor* has great advice on how to make these documents shine. Create both HTML and PDF versions of these primary documents. In the cover letter, use **bold face** on the names of faculty member. Keep your cover letter very brief—two paragraphs at most. Have local faculty outside your area read over your materials, and most important of all, listen to their feedback. List impact factors or acceptance rates alongside your publications in your CV. This gives people outside your field a crude metric to judge your publications.

2. Social media profiles. Employers likely will search for your name, and they do anticipate you'll have a Facebook page, LinkedIn account, and so forth. It's okay to have online profiles, but clean up your image and politically sterilize them before your job search even begins.

3. Web site. You'll need a straightforward and professional web site to host your primary documents and all your published works or links to them if they are restricted.

While I was searching for a job, I tracked my web site's traffic and found that 69 percent of visitors originating from academic institutions hit my homepage, whereas only 18 percent looked at my CV. Ten percent got to my research statement, while only 3 percent hit my teaching statement. The take-away? Your homepage is more important than even your CV.

It's also helpful to know that interviewers will usually wait until the last minute to download and print paper copies of your materials, perhaps the same day you arrive for an interview. Keep everything current.

Once these application materials land you a few interviews, you'll still have to prepare a job talk, your answers to the standard questions that come up during the typical academic job interview, and your approach to negotiating and discussing start-up packages.

For more advice on preparing for your academic job search, see the related articles on http://xrds.acm.org under Resources.

—*Matt Might*

"It's okay to have online profiles, but clean up your image and politically sterilize them before your job search even begins."

1.6 Petaflops

Speed of the fastest super computer; futurist Ray Kurzweil believes the human brain has a power of 10 petaflops.*

*http://www.wired.com/wiredscience/2008/11/supercomputers/

5 billion

Number of words that have been digitized using ReCaptcha.

UPDATES

Establishing an ACM Student Chapter
Activity Ideas for University Groups

The previous "Updates" column (Fall 2010), introduced two ACM chapters and highlighted some interesting activities they've hosted. Since then, readers have asked for more information related to starting a new chapter, specifically, "What will the chapter do?"

Most chapters organize workshops, tutorials, lectures, and set up a web site. But there are a lot of innovative and exciting ideas we'd like to share.

Shadowing. High school students get paired with college students and spend the day attending classes and socializes. High school students gain the opportunity to learn more about a college computer science major, since they may not have the opportunity to at their schools.

Mock interviews. Professors or professionals from area businesses volunteer to participate in mock interviews for students contemplating graduate school or entering the work world.

Promote CS with brain games. Post flyers with brain games and puzzles that exercise one's problem-solving skills (see page 60). Add a line that reads, "If you like these puzzles, you might like computer science, too!"

ACM student chapters teach programming to non-CS students.

To get more ideas, I spoke to other ACM student members in Athens and North Carolina.

ATHENS, GREECE

Under a mainly business environment, a very young chapter in the Economics and Business school of Athens University tried to attract new undergraduate students to join this fall. They wanted more people to get involved in an unconstrained and, at the same time, practical way.

So they introduced the buddy system.

"We believe that the buddy system manifests exactly the point of being part of something bigger, expressing ourself and collaborating with each other," Chris Andreikos, vice-chair of the chapter tells us.

In the buddy system, students pair up (or form small groups) and set up times to meet outside the classroom throughout the semester. Younger students are paired with older students who share their interests and hobbies, and they spend time together as mentor and mentee. Introduce a faculty member or a CS professional, and there it is: tri-mentoring.

WILMINGTON, NORTH CAROLINA, USA

The University of North Carolina-Wilmington ACM Student Chapter has seen a sharp increase in the number of active members during the last years, during which they hosted simple 20-minute talks where students could get public speaking experience and talk about some subject of interest in computing. Due to the popularity of the events, they have created a "student lecture" series. It's similar to a workshop, where volunteers from the computer science major teach other students the basics of certain software programs. In this case, C# is being taught through the use of the Xbox 360 and XNA.

Ben Barbour, the chapter president and a professional C# developer for nearly 10 years, says, "I was thinking about ways to engage not only ACM members but other students across the program and thought that teaching a programming language in an environment that most every student is very familiar with would help spark ideas and motivate students to experiment and learn on their own."

More ideas can be found at http://www.acm.org/chapters/students/Ideas.pdf. Let us know what your chapter is doing by emailing xrds@acm.org.

—*Vaggelis Giannikas*

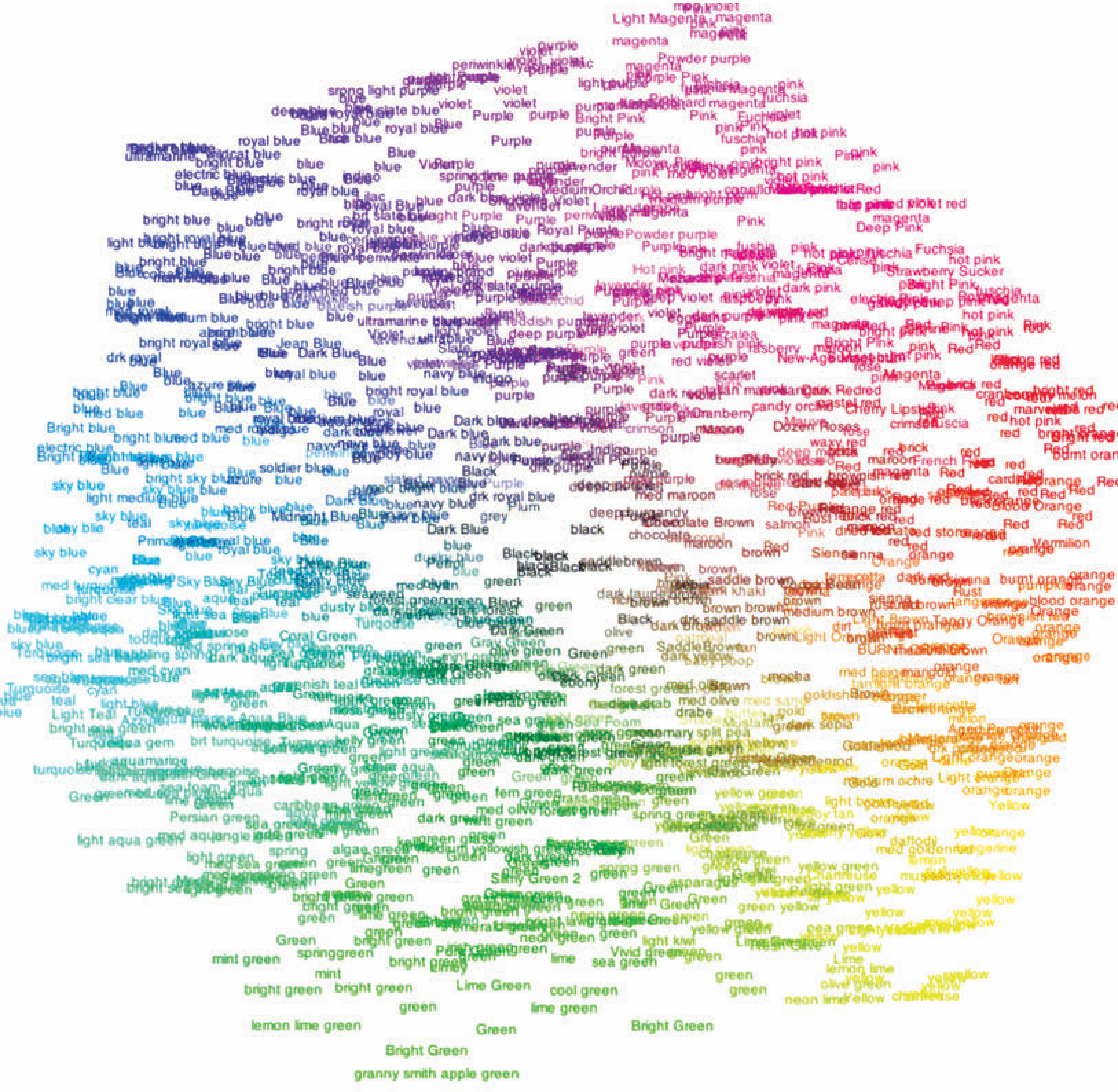


Figure 1: We showed thousands of random colors to people on Amazon Mechanical Turk and asked what they would call them. This image contains about 1,300 colors and the names people gave them. Each is printed in its color and positioned on a color wheel.

Massive Multiplayer Human Computation for Fun, Money, and Survival

Labor-on-demand—it's like cloud computing but with human workers.

By Lukas Biewald

DOI: 10.1145/1869086.1869093

Before the internet enabled human beings to connect as they do now, collecting large-scale datasets that require human computation was a time-consuming and expensive process. At CrowdFlower, we produce new datasets on-demand by routing tasks to large groups of distributed workers who work simultaneously. We see people collecting creative and innovative datasets for businesses, for fun, and even to improve the lives of others.

DEMOCRATIZING DATA COLLECTION

Topics of study in disciplines that focus on quantitative or technical data, like machine-learning research, have always been limited by the availability of datasets. For example, the Brown Corpus is a dataset compiled in the 1960s that has served as the basis for thousands of linguistics studies. It has been exhaustively parsed and tabbed. Graduate students would center entire research plans on the availability of previously collected data, and as a result, generations of papers on word disambiguation were tailored to the constraints of old data.

Crowdsourcing democratizes the data-collection process, cutting researchers' reliance on stagnant, overused datasets. Now, anyone can gather data overnight rather than waiting years. However, some of the data collection may be sloppy. CrowdFlower addresses this issue by building robust quality-control mechanisms in order to standardize the results that come back from the crowd. The type of crowd, task design, and quality control tactics all affect the quality of the data. The important thing to remember is that crowdsourcing provides channels that allow researchers, businesses, or even armchair social scientists to gather data — having high quality data obviously affects the accuracy of the research.

MY CROWDSOURCING ROOTS

The first time I used Amazon's Mechanical Turk was at a search-engine startup, Powerset (http://powerset.com, later acquired by Microsoft). I used Mechanical Turk to compare the quality of our search-relevancy algorithm against Yahoo! and Google. Initially, I thought it would be necessary to hire a team of people to compare the quality of results every day over

"Crowdsourcing democratizes the data-collection process, cutting researchers' reliance on stagnant, overused datasets. Now, anyone can gather data overnight rather than waiting years."

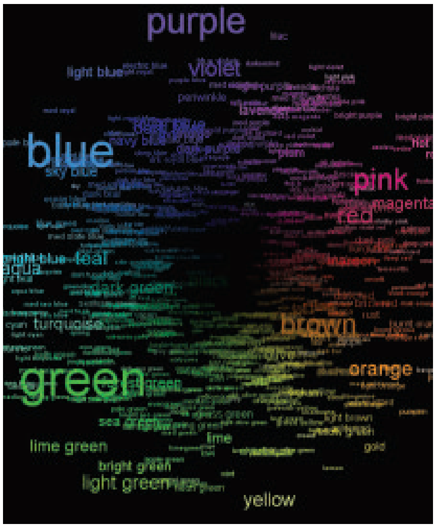


Figure 2: Martin Wattenberg's cloud view of 10,000 points of color data were gathered by CrowdFlower through Amazon Mechanical Turk.

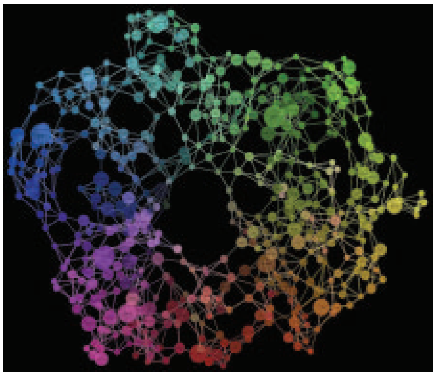


Figure 3: David Sparks' network and cluster diagram of the 10,000 points of color data were gathered by CrowdFlower through Amazon Mechanical Turk.

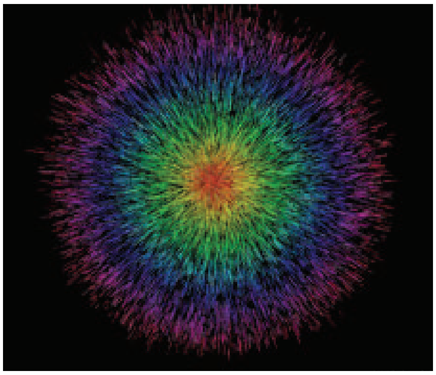


Figure 4: Chris Harrison's flower image comprised the names of 10,000 points of color data, collected by CrowdFlower using Amazon Mechanical Turk.

the course of months. Instead, I set up an experimental task with no quality control, put in about $50, and let it run overnight. The data that came back was noisy, but I was able to find meaningful differences between the search engines.

The following experiments showcase the role that crowdsourcing plays in data collection. Many of these are featured on our blog, and we often post new datasets and projects. We invite researchers to post new experiments on our site.

WHERE DOES BLUE END AND RED BEGIN?

One of the first experiments we ran used a design similar to Brent Berlin and Paul Kay's World Color Survey (http://www.icsi.berkeley.edu/wcs/, itself an expansion of the experiment conducted in Berlin and Kay's 1969 *Basic Color Terms*). This survey tested the theory that different cultures share inherent constants in how they identify colors, provided that their languages are at similar stages of development. To conduct the survey, they showed color patches to speakers of many different languages and asked for the names of each color.

In our experiment, we showed 10,000 random patches of color to people on Mechanical Turk in March 2008. In a free-text response, we instructed participants to label each color using one name only. We collected data from 238 different individuals (see **Figure 1**). Some of the more creative answers included light-cordovan brown, maritime blue, and prairie grass.

The data collection was not rigorous, but we still had a ton of data — more than the original survey. We released it to the public, so others could access it and conduct their own experiments. We were amazed by how many people used the dataset. Note the following four examples.

Scientist and data visualization artist Martin Wattenberg used the data to build a visualization (see **Figure 2**) by grouping together identical names, taking an average position, and sizing the words by frequency. Jeff Clark riffed on this design and wrote a 3D explorer that allows users to fly around a space of the color labels; see http://www.neoformix.com/2008/ColorNamesExplorer.html.

David Sparks, a graduate student in Duke University's Political Science department, used the data to create network and cluster diagrams like the one in **Figure 3**, which was computed from a similarity metric on color names. The size of a node corresponds to the label's frequency.

Last, but not least, Chris Harrison, a PhD candidate at Carnegie Mellon University (and editor-in-chief of this magazine), combined our data with results from his own previous experiment, and created intricate flower and spiral images, as seen in **Figure 4**, that comprise the color names.

BEAUTY AND BEHOLDERS

We used to host a site called FaceStat which allowed people to upload pictures of themselves and get strangers' first impressions. In less than a year, we used the site to collect more than 10 million multivariate judgments on more than 100,000 faces. We tracked and plotted how age (or perceived age) affects a person's perceived attractiveness, trustworthiness, intelligence, and political beliefs (see **Figure 5**). Aggregating millions of these snap decisions told us a lot about our own biases in surprising ways. Here are a few of our findings:

You might think that 20-year-olds would be judged as most attractive, but that was not the case. The top three ages that participants found most "attractive" were (ranked in order of attractiveness): babies, 26-year-olds, and people over the age of 60. In fact, the closer a subject got to the "peak" age of 18, the less attractive they were judged to be (see **Figure 6**).

Women are judged as much more trustworthy than men, with the lowest trust scores for adolescent males. (The latter observation may be unsurprising, to the chagrin of upstanding teenage males). Interestingly, there is a large jump in trustworthiness for both men and women between 20 and 30, and between 50 and 60. Children and old people are judged as more intelligent, with males in their 20s getting the lowest scores. (Again, young men just can't catch a break.)

Finally, as men and women get older they are thought to be more conservative. Although young women are perceived as more liberal than young men, that gap disappears after 25.

The raw data is available at http://data.doloreslabs.com, and O'Reilly's *Beautiful Data* features the full experiment and useful hacks for processing data.

CROWDSOURCING BENFORD'S LAW

If you asked someone to pick a number, what would it be? What's the likelihood that it would begin with one? How about seven, or two? Benford's Law states that about 30 percent of the leading digits will be ones, 17.6 percent will be twos, and so on down to around 4.6 percent of the numbers beginning with nine.

Because we had easy access to a large pool of people, we tested the law by asking subjects to pick any number greater than zero. In a little over an hour, we received 500 valid results that roughly corresponded to the law, as you can see in **Figure 7**. However, some laws were made to be broken, and we did notice a slight departure from the law when it came to people who chose numbers beginning with four and five. While 42 subjects chose a number beginning with four, 51 subjects chose a number beginning with five.

CROWDSOURCING ETHICS

Crowds can also be used to source answers to philosophical questions. Stalin said, "A single death is a tragedy; a million deaths is a statistic." So what about 100 deaths? What about five? We tested this experimentally by asking people on Amazon Mechanical Turk to participate in the classic philosophical conundrum "The Trolley Problem," in which a person must decide whether to sacrifice one person in order to save several others.

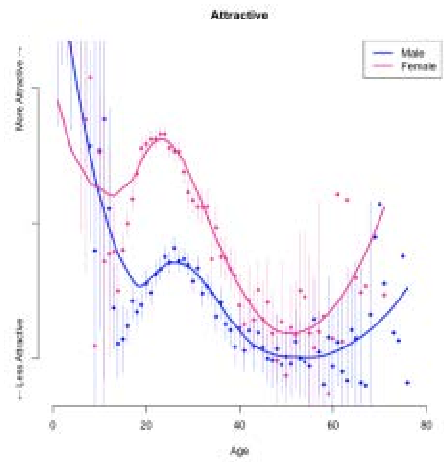


Figure 6: The graph above shows attractiveness based on subjects' perceived age.

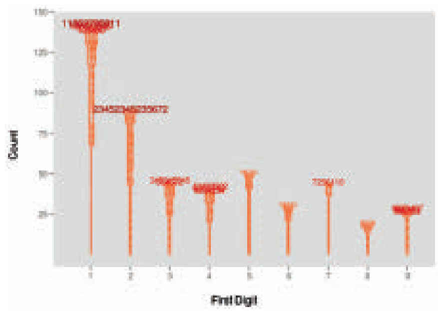


Figure 7: University of California-Berkeley PhD candidate Aaron Shaw plotted this data showing the results from 500 workers who were asked to pick a single digit greater than zero. All the numbers are sorted by magnitude and leading digit. The height captures the frequency of that leading digit. The size of the numerals corresponds to the magnitude of the number, and the color corresponds roughly to its order of magnitude (red=big). Within each digit, the numbers are sorted along the *y*-axis.

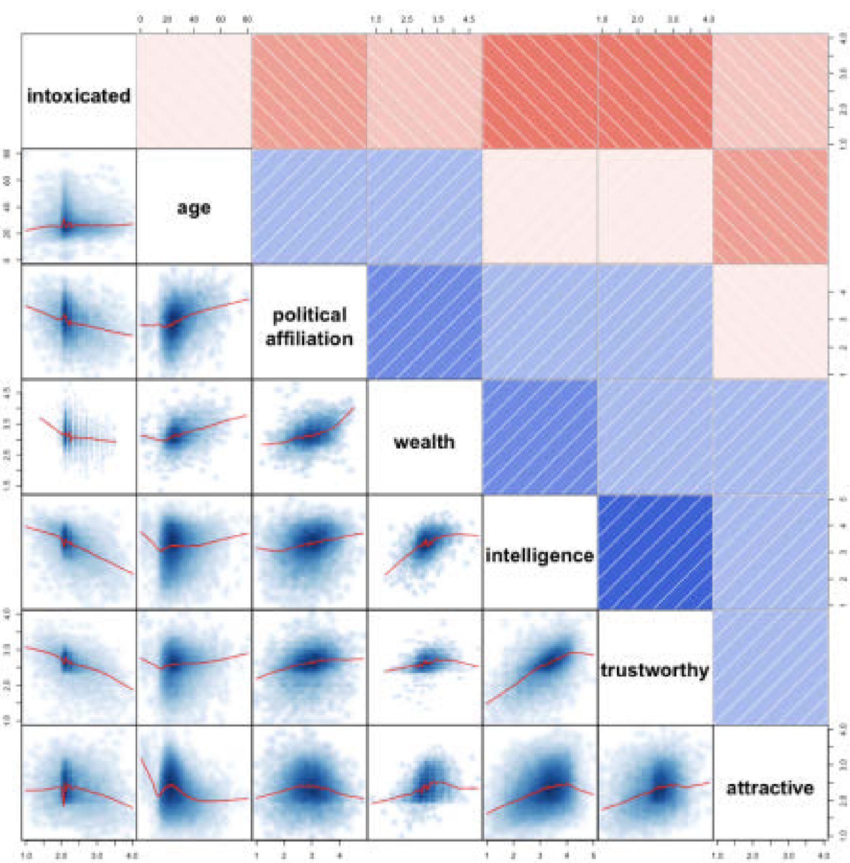


Figure 5: A scatterplot matrix of the attributes: Every pair of attributes has two graph panels. The bottom-left panels are simply *x-y* scatterplot with trendlines for each attribute pair. The top-right panels show corrgrams of Pearson correlations: blue=positively correlated attributes, and red=negatively correlated ones. Looking at the attributes near the center, we see that conservatism, wealth, intelligence, and trustworthiness are all positively correlated with each other, hence the dark blue squares. Attractiveness is more complex. It tends to show a peak in the middle of the graph, especially with intoxication, age, and political affiliation.

The three sample scenarios are below. Here, the scenarios refer to saving five people, but we varied the number of people saved between 1 and 1,000 to see if it would affect the results.

Scenario A. A trolley is running out of control down a track. In its path are five people who have been tied to the track. Fortunately, you can flip a switch, which will lead the trolley down a different track to safety. Unfortunately, there is a single person tied to that track. Should you flip the switch?

Scenario B. As before, a trolley is hurtling down a track toward five people. You are on a bridge under which it will pass, and you can stop it if you drop something heavy in its way. As it happens, there is a very fat man next to you—your only way to stop the trolley is to push him over the bridge and onto the track, killing him to save five. Should you proceed?

Scenario C. A brilliant transplant surgeon has five patients, each in need of a different organ, each of whom will die without that organ. Unfortunately, there are no organs available to perform any of these five transplant operations. A healthy young traveler, just passing through the city the doctor works in, comes in for a routine checkup. In the course of doing the checkup, the doctor discovers that his organs are compatible with all five of his dying patients. Suppose further that if the young man were to disappear, no one would suspect the doctor. Should the doctor sacrifice the man to save his other patients?

How does our decision change based on the number of people who will die? The results were unexpected. For all three scenarios, subjects were increasingly willing to kill one person when it meant saving up to 100 people, but this willingness dipped when it meant saving between 100 and 200 people, and began to rise again when more than 200 people could be saved.

Each scenario also affected subjects' ethical calculus. Subjects were more willing to sacrifice a life when they controlled the trolley switch (Scenario A) than when someone else (i.e., the surgeon in Scenario C) acted as executioner (see **Figure 8**). By putting this experiment out to the crowd, we were able to gather responses from more than 100 subjects in a matter of hours.

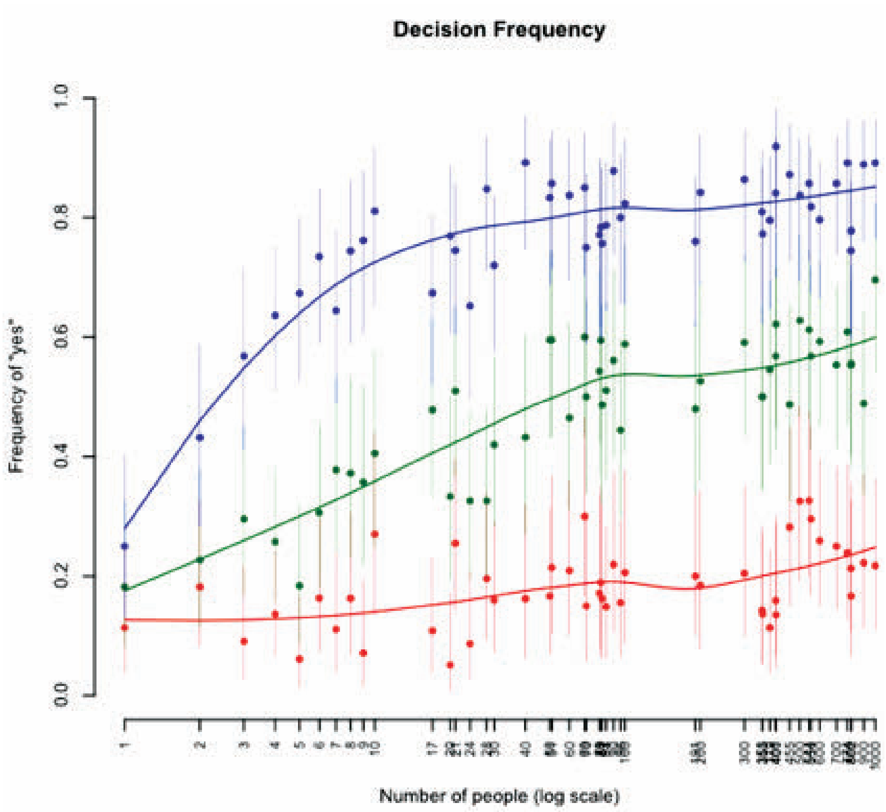


Figure 8: The frequencies of responses for three different variations of the Trolley Problem (plotted on a log scale with a LOESS fit).

CROWDSOURCING BUSINESS

CrowdFlower provides a way for businesses to complete large tasks by giving them access to tens of thousands or hundreds of thousands of workers. By layering our technology across multiple labor channels, CrowdFlower enables access to a large sample of people. This means businesses can connect to a labor force in the cloud depending on their needs.

Just as cloud computing eliminated the need for businesses to predict how many servers they were going to need at any given time, crowdsourced labor eliminates the need for businesses to predict how many people they're going to need at any given minute. The quality-control algorithms we've developed ensure that workers are doing the tasks well, that they are trained when they make mistakes, and that they are removed if they are inefficient or spammy.

CrowdFlower has been able to assist clients in many ways, such as improving the accuracy of data for clients who publish business listings: e.g., business leader information (CEO and CFO names), contact information (phone, fax, and address), and company information (company name, description, and industry label).

In particular, our use of quality-controlled human intelligence excels at removing duplicate business listings that a computer program would gloss over. For example, a computer cannot say for certain whether the McDonald's restaurant in Manhattan and the McDonald's restaurant in New York City are the same place, but a person could do the additional research required to confirm whether the two restaurants are the same. In one instance, we improved the accuracy of a client's data by more than 20 percent.

> **“Crowdsourcing provides channels that allow researchers, businesses, or even armchair social scientists to gather data.”**

THE GREATER GOOD

Crowdsourced data collection will continue to benefit businesses and revolutionize academic research. It can also benefit disenfranchised people by giving them access to dignified work.

For starters, the microtasks involved in crowdsourced labor mean that anyone anywhere can be productive, even if just for 10 minutes. They don't have to kill time playing solitaire at their computers or working on crosswords. Instead, they could be paid to do a job, earn virtual credits for an online game, or even give work to people in developing nations through a paired work program.

The developing world needs employment opportunities that crowdsourcing can provide by connecting women, youth, and refugees living in poverty to computer-based work. One of CrowdFlower's labor channels, Samasource, does just that.

CrowdFlower and Samasource created GiveWork, an iPhone app (see **Figure 9**) that allows users to support people in developing countries by completing short, on-screen tasks, which can either give a donation or an additional unit of work that is used for training purposes. People volunteer to tag a video or trace a road alongside someone who is learning computer skills. This is just the beginning for mobile-based crowdsourced labor.

Further, getting work to refugee populations is difficult whenever that work requires raw materials: e.g., building physical structures; but projects that require building information can move quickly and globally.

Crowdsourced labor has begun to level the playing field with respect to job access. This lends greater meritocracy to the job market: it is a natural extension of what the Internet has already done, but rather than "who you know," the focus is on "what you know." Thus, a person in Berlin or Jakarta is not immediately ruled out of a job due geography.

DISASTER RELIEF

Crowdsourcing also helps disaster relief efforts because disasters are, by definition, unpredictable, and relief requires a scalable workforce. The response to the Haitian earthquake in January 2010 demonstrated how a rapidly deployed workforce of far-flung volunteers can be critical to emergency relief efforts. After the quake, aid workers flooded Port-au-Prince, but they lacked information about who needed help, where the victims were, and what type of help they needed. FrontLineSMS and the U.S. Department of State worked with Haitian telcom companies to set up an SMS short code, allowing Haitians to submit real-time reports using less bandwidth than the two-way audio that had caused system outages on the country's cell networks. The messages that came in were in Kréyol and the aid agencies were unable to translate the messages fast enough.

CrowdFlower provided the infrastructure to route SMS texts to hundreds of thousands of Haitians (located by Samasource) who translated texts from Port-au-Prince in real time and categorized the victims' issues, allowing the agencies to direct specialists to the people who needed their services: e.g., getting potable water to thirsty people, routing doctors to injured people. Further, so that agencies could see hotspots, maps were created through Ushahidi, an open-source platform that allows people or organizations to collect and visualize information (see **Figure 10**).

The rapid proliferation of broadband, wireless, and cell phone technology has revolutionized disaster relief efforts so that now, anyone with a computer or phone can provide assistance.

THE FUTURE OF DATA DEMOCRACY

What does the future look like for crowdsourcing, human computation, data exchange, and data transparency? When data is made widely available, it becomes widely analyzable, and through this process, crowdsourcing can empower us all.

Figure 9: The iPhone app GiveWork lets users complete small tasks and donate their payment to developing nations.

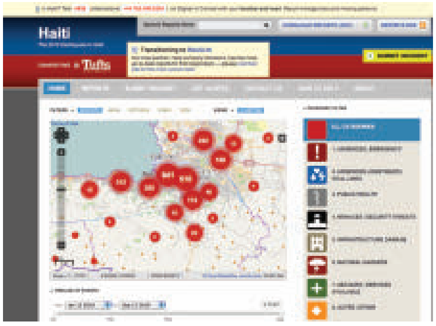


Figure 10: Ushahidi is a data-mapping and visualization platform used in the relief efforts following the earthquake in Haiti in 2009.

Biography

Lukas Biewald is the CEO of CrowdFlower. Prior to co-founding CrowdFlower, he was a senior scientist and manager within the ranking and management team at Powerset, Inc., a natural language search technology company later acquired by Microsoft. Biewald has also led the search relevance team for Yahoo! Japan. He graduated from Stanford University with a BS in mathematics and an MS in computer science. He is an expert-level Go player.

Analyzing the Amazon Mechanical Turk Marketplace

An associate professor at New York University's Stern School of Business uncovers answers about who the workers are in paid crowdsourcing and how much they earn.

By Panagiotis G. Ipeirotis

DOI: 10.1145/1869086.1869094

Amazon Mechanical Turk (AMT) is a popular crowdsourcing marketplace, introduced by Amazon in 2005. The marketplace is named after an 18th century "automatic" chess-playing machine, which was handily beating humans in chess games. Of course, the robot was not using any artificial intelligence algorithms back then. The secret of the Mechanical Turk machine was a human operator, hidden inside the machine, who was the real intelligence source.

AMT is also a marketplace for small tasks that cannot be easily automated. For example, humans can tell if two different descriptions correspond to the same product, can easily tag an image with descriptions of its content, or can easily transcribe with high quality an audio snippet—though all those tasks are extremely difficult for computers to do.

Using Mechanical Turk, computers can use a programmable API to post tasks on the marketplace, which are then fulfilled by human users. This API-based interaction gives the impression that the task can be automatically fulfilled, hence the name.

In the marketplace, employers are known as requesters and they post tasks, called human intelligence tasks, or HITs. The HITs are then picked up by online users, referred to as workers, who complete them in exchange for a small payment, typically a few cents per HIT.

Since the concept of crowdsourcing is relatively new, many potential participants have questions about the AMT marketplace. For example, a common set of questions that pop up in an "introduction to crowdsourcing and AMT" session are the following:

- Who are the workers that complete these tasks?
- What type of tasks can be completed in the marketplace?
- How much does it cost?
- How fast can I get results back?
- How big is the AMT marketplace?

For the first question, about the demographics of the workers, past research [5, 6] indicated that the workers that participate on the marketplace are mainly coming from the United States, with an increasing proportion coming from India. In general, the workers are representative of the general Internet user population but are generally younger and, correspondingly, have lower incomes and smaller families.

"By observing the practices of the successful requesters, we can learn more about what makes crowdsourcing successful, and increase the demand from the smaller requesters."

At the same time, the answers for the other questions remain largely anecdotal and based on personal observations and experiences. To understand better what types of tasks are being completed today using crowdsourcing techniques, we started collecting data about the marketplace. Here, we present a preliminary analysis of the findings and provide directions for interesting future research.

DATA COLLECTION

We started gathering data about AMT in January 2009, and we continue to collect data at the time of this writing. Every hour, we crawl the list of HITs available on AMT and keep the status of each available HIT group (groupid, requester, title, description, keywords, rewards, number of HITs available within the HIT group, qualifications required and time of expiration). We also store the HTML content of each HIT.

Following this approach, we could find the new HITs being posted over time, the completion rate of each HIT, and the time that they disappear from the market because they have either been completed or expired, or because a requester canceled and removed the remaining HITs from the market. (Identifying expired HITs is easy, as we know the expiration time of a HIT. Identifying cancelled HITs is a little trickier. We need to monitor the usual completion rate of a HIT over time and see if it is likely, at the time of disappearance, for the remaining HITs to have been completed within the time since the last crawl.)

A shortcoming of this approach is that it cannot measure the redundancy of the posted HITs. So, if a single HIT needs to be completed by multiple workers, we can only observe it as a single HIT.

The data are also publicly available through the website http://www.mturk-tracker.com [1].

From January 2009 through April 2010, we collected 165,368 HIT groups, with 6,701,406 HITs total, from 9,436 requesters. The total value of the posted HITs was $529,259. These numbers, of course, do not account for the redundancy of the posted HITs, or for HITs that were posted and disappeared between our crawls. Nevertheless, they should be good approximations (within an order of magnitude) of the activity of the marketplace.

TOP REQUESTERS AND FREQUENTLY POSTED TASKS

One way to understand what types of tasks are being completed in the marketplace is to find the "top" requesters and analyze the HITs that they post. **Table 1** shows the top requesters, based on the total rewards of the HITs posted, filtering out requesters that were active only for a short period of time.

Table 1: Top Requesters based on the total posted rewards available to a single worker (January 2009–April 2010).

Requester ID	Requester Name	#HIT groups	Total HITs	Rewards	Type of tasks
A3MI6MIUNWCR7F	CastingWords	48,934	73,621	$59,099	Transcription
A2IR7ETVOIULZU	Dolores Labs	1,676	320,543	$26,919	Mediator for other requesters
A2XL3J4NH6JI12	ContentGalore	1,150	23,728	$19,375	Content generation
A1197OGL0WOQ3G	Smartsheet.com Clients	1,407	181,620	$17,086	Mediator for other requesters
AGW2H4I480ZX1	Paul Pullen	6,842	161,535	$11,186	Content rewriting
A1CTI3ZAWTR5AZ	Classify This	228	484,369	$9,685	Object classification
A1AQ7EJ5P7ME65	Dave	2,249	7,059	$6,448	Transcription
AD7C0BZNKYGYV	QuestionSwami	798	10,980	$2,867	Content generation and evaluation
AD14NALRDOSN9	retaildata	113	158,206	$2,118	Object classification
A2RFHBFTZHX7UN	ContentSpooling.net	555	622	$987	Content generation and evaluation
A1DEBE1WPE6JFO	Joel Harvey	707	707	$899	Transcription
A29XDCTJMAE5RU	Raphael Mudge	748	2,358	$548	Website feedback

We can see that there are very few active requesters that post a significant amount of tasks in the marketplace and account for a large fraction of the posted rewards. Following our measurements, the top requesters listed in Table 1 (which is 0.1 percent of the total requesters in our dataset), account for more than 30 percent of the overall activity of the market.

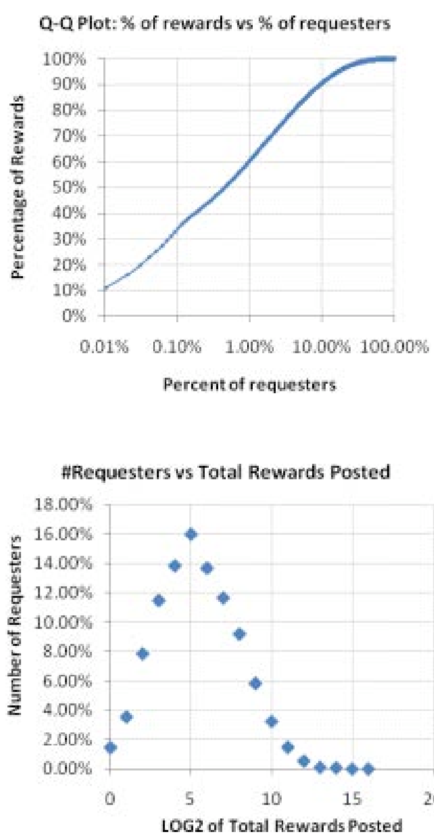
Given the high concentration of the market, the type of tasks posted by the requesters shows the type of tasks that are being completed in the marketplace. Castingwords is the major requester, posting transcription tasks frequently. There are also two other semi-anonymous requesters posting transcription tasks as well.

Among the top requesters we also see two mediator services, Dolores Labs (aka Crowdflower) and Smartsheet.com, who post tasks on Mechanical Turk on behalf of their clients. Such services are essentially aggregators of tasks, and provide quality assurance services on top of Mechanical Turk. The fact that they account for approximately 10 percent of the market indicates that many users that are interested in crowdsourcing prefer to use an intermediary that address the concerns about worker quality, and also allow posting of complex tasks without the need for programming.

We also see that four of the top requesters use Mechanical Turk in order to create a variety of original content, from product reviews, feature stories, blog posts, and so on. (One requester, "Paul Pullen," uses Mechanical Turk to paraphrase existing content, instead of asking the workers to create content from scratch.) Finally, we see that two requesters use Mechanical Turk in order to classify a variety of objects into categories. This was the original task for which Mechanical Turk was used by Amazon.

The high concentration of the market is not unusual for any online community. There is always a long tail of participants that has significantly lower activity than the top contributors. **Figure 1** shows how this activity is distributed, according to the value of the HITs posted by each requester. The *x*-axis shows the log2 of the value of the posted HITs and the *y*-axis shows what percentage of requesters has this level of activity. As we can see, the dis-

Figure 1: Number of requesters vs. total rewards posted.



tribution is approximately log-normal. Interestingly enough, this is approximately the same level of activity demonstrated by workers [5].

For our analysis, we wanted to also examine the marketplace as a whole, to see if the HITs submitted by other requesters were significantly different than the ones posted by the top requesters. For this, we measured the popularity of the keywords in the different HITgroups, measuring the number of HITgroups with a given keywords, the number of HITs, and the total amount of rewards associated with this keyword. **Table 2** shows the results.

Our keyword analysis of all HITs in our dataset indicates that transcription is indeed a very common task on the AMT marketplace. Notice that it is one of the most "rewarding" keywords and appears in many HITgroups, but not in many HITs. This means that most of the transcription HITs are posted as single HITs and not as groups of many similar HITs. By doing a comparison of the prices for the transcription HITs, we also noticed that it is a task for which the payment per HIT is comparatively high. It is unclear at this point if this is due to the high expectation for quality or whether the higher price simply reflects the higher effort required to complete the work.

Beyond transcription, **Table 2** indicates that classification and categorization are indeed tasks that appear in many (inexpensive) HITs. **Table 2** also indicates that many tasks are about data collection, image tagging and classification, and also ask workers for feedback and advice for a variety of tasks (e.g., usability testing of websites).

PRICE DISTRIBUTIONS

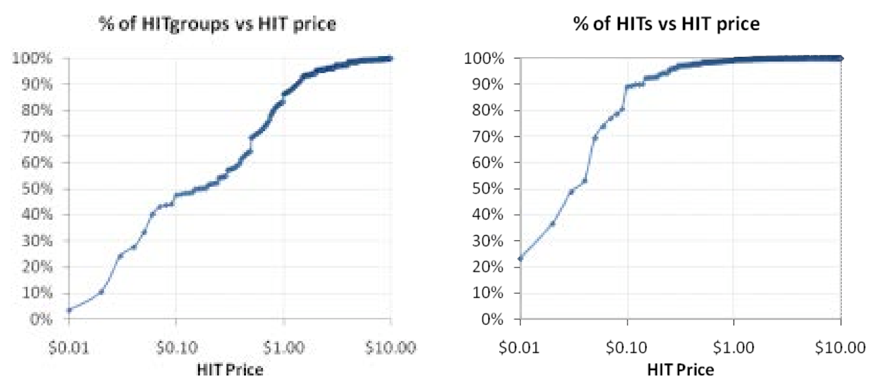
To understand better the typical prices paid for crowdsourcing tasks on AMT, we examined the distribution of the HIT prices and the size of the posted HITs. **Figure 2** illustrates the results. When examining HIT groups, we can see that only 10 percent have a price tag of $0.02 or less, 50 percent of the HITs have price above $0.10, and 15 percent of the HITs come with a price tag of $1 or more.

However, this analysis can be misleading. In general, HITgroups with a

Table 2: The top 50 most frequent HIT keywords in the dataset, ranked by total reward amount, number of HITgroups, and number of HITs.

Keyword	Rewards	Keyword	#HITGroups	Keyword	#HITs
data	$192,513	castingwords	48,982	product	4,665,449
collection	$154,680	cw	48,981	data	3,559,495
easy	$93,293	podcast	47,251	categorization	3,203,470
writing	$91,930	transcribe	40,697	shopping	3,086,966
transcribe	$81,416	english	34,532	merchandise	2,825,926
english	$78,344	mp	33,649	collection	2,599,915
quick	$75,755	writing	29,229	easy	2,255,757
product	$66,726	question	21,274	categorize	2,047,071
cw	$66,486	answer	20,315	quick	1,852,027
castingwords	$66,111	opinion	15,407	website	1,762,722
podcast	$64,418	short	15,283	category	1,683,644
mp	$64,162	advice	14,198	image	1,588,586
website	$60,527	easy	11,420	search	1,456,029
search	$57,578	article	10,909	fast	1,372,469
image	$55,013	edit	9,451	shopzilla	1,281,459
builder	$53,443	research	9,225	tagging	1,028,802
mobmerge	$53,431	quick	8,282	cloudsort	1,018,455
write	$52,188	survey	8,265	classify	1,007,173
listings	$48,853	editing	7,854	listings	962,009
article	$48,377	data	7,548	tag	956,622
research	$48,301	rewriting	7,200	photo	872,983
shopping	$48,086	write	7,145	pageview	862,567
categorization	$44,439	paul	6,845	this	845,485
simple	$43,460	pullen	6,843	simple	800,573
fast	$40,330	snippet	6,831	builder	796,305
categorize	$38,705	confirm	6,543	mobmerge	796,262
email	$32,989	grade	6,515	picture	743,214
merchandise	$32,237	sentence	6,275	url	739,049
url	$31,819	fast	5,620	am	613,744
tagging	$30,110	collection	5,136	retail	601,714
web	$29,309	review	4,883	web	584,152
photo	$28,771	nanonano	4,358	writing	548,111
review	$28,707	dinkle	4,358	research	511,194
content	$28,319	multiconfirmsnippet	4,218	email	487,560
articles	$27,841	website	4,140	v	427,138
category	$26,656	money	4,085	different	425,333
flower	$26,131	transcription	3,852	entry	410,703
labs	$26,117	articles	3,540	relevance	400,347
crowd	$26,117	search	3,488	flower	339,216
doloreslabs	$26,117	blog	3,406	labs	339,185
crowdflower	$26,117	and	3,360	crowd	339,184
delores	$26,117	simple	3,164	crowdflower	339,184
dolores	$26,117	answers	2,637	doloreslabs	339,184
deloreslabs	$26,117	improve	2,632	delores	339,184
entry	$25,644	retranscribe	2,620	dolores	339,184
tag	$25,228	writer	2,355	deloreslabs	339,184
video	$25,100	image	2,322	find	338,728
editing	$24,791	confirmsnippet	2,291	contact	324,510
classify	$24,054	confirmtranscription	2,288	address	323,918
answer	$23,856	voicemail	2,202	editing	321,059

Figure 2: Distribution of HITgroups and HITs according to HIT Price.



high price only contain a single HIT, while the HITgroups with large number of HITs have a low price. Therefore, if we compute the distribution of HITs (not HITgroups) according to the price, we can see that 25 percent of the HITs created on Mechanical Turk have a price tag of just $0.01, 70 percent have a reward of $0.05 or less, and 90 percent pay less than $0.10. This analysis confirms the common feeling that most of the tasks on Mechanical Turk have tiny rewards.

Of course, this analysis simply scratches the surface of the bigger problem: How can we automatically price tasks, taking into consideration the nature of the task, the existing competition, the expected activity level of the workers, the desired completion time, the tenure and prior activity of the requester, and many other factors? How much should we pay for an image tagging task, for 100,000 images in order to get it done within 24 hours? Building such models will allow the execution of crowdsourcing tasks to become easier for people that simply want to "get things done" and do not want to tune and micro-optimize their crowdsourcing process.

POSTING AND SERVING PROCESSES

What is the typical activity in the AMT marketplace? What is the volume of the transactions? These are very common questions from people who are interested in understanding the size of the market and its demonstrated capacity for handling big tasks. (Detecting the true capacity of the market is a more involved task than simply measuring its current serving rate. Many workers may show up only when there is a significant amount of work for them, and be dormant under normal loads.)

One way to approach such questions is to examine the task posting and task completion activity on AMT. By studying the posting activity we can understand the demand for crowdsourcing, and the completion rate shows how fast the market can handle the demand. To study these processes, we computed, for each day, the value of tasks being posted by AMT requesters and the value of the tasks that got completed in each day.

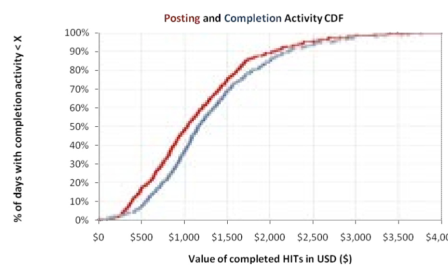
We present first an analysis of the two processes (posting and completion), ignoring any dependencies on task-specific and time-specific factors. **Figure 3** illustrates the distributions of the posting and completion processes. The two distributions are similar but we see that, in general, the rate of completion is slightly higher than the rate of arrival. This is not surprising and is a required stability condition. If the completion rate was lower than the arrival rate, then the number of incomplete tasks in the marketplace would go to infinity.

We observed that the median arrival rate is $1,040 per day and the median completion rate is $1,155 per day. If we assume that the AMT marketplace behaves like an M/M/1 queuing system, and using basic queuing theory, we can see that a task worth $1 has an average completion time of 12.5 minutes, resulting in an effective hourly wage of $4.80.

Of course, this analysis is an oversimplification of the actual process. The tasks are not completed in a first-in, first-out manner, and the completion rate is not independent of the arrival rate. In reality, workers pick tasks following personal preferences or by the AMT interface. For example Chilton *et al*. [4] indicated that most workers use two of the main task sorting mechanisms provided by AMT to find and complete tasks ("recently posted" and "largest number of HITs" orders). Furthermore, the completion rate is not independent of the arrival rate.

When there are many tasks available, more workers come to complete tasks, as there are more opportunities to find and work for bigger tasks, as opposed to working for one-time HITs. As a simple example, consider the dependency of posting and completion rates on the day of the week. Figure 4 illustrates the results.

Figure 3: The distribution of the arrival and completion rate on the AMT marketplace, as a function of the USD ($) value of the posted/completed HITs.



The posting activity from the requesters is significantly lower over the weekends and is typically maximized on Tuesdays. This can be rather easily explained. Since most requesters are corporations and organizations, most of the tasks are being posted during normal working days. However, the same does not hold for workers. The completion activity is rather unaffected by the weekends. The only day on which the completion rate drops is on Monday, and this is most probably a side-effect of the lower posting rate over the weekends. (There are fewer tasks available for completion on Monday, due to the lower posting rate over the weekend.)

An interesting open question is to understand better how to model the marketplace. Work on queuing theory for modeling call centers is related and can help us understand better the dynamics of the market and the way that workers handle the posted tasks. Next, we present some evidence that modeling can help us understand better the shortcomings of the market and point to potential design improvements.

COMPLETION TIME DISTRIBUTION

Given that the system does not satisfy the usual queuing assumptions of M/M/1 [7] for the analysis of completion times, we analyzed empirically the completion time for the posted tasks. The goal of this analysis was to understand what approaches may be appropriate for modeling the behavior of the AMT marketplace.

Our analysis indicated that the completion time follows (approximately) a power law, as illustrated in **Figure 5**. We observe some irregularities, with some outliers at approximately 12 hours and at the seven-day completion times. These are common "expiration times" set for many HITs, hence the sudden disappearance of many HITs at that point. Similarly, we see a different behavior of HITs that are available for longer than one week: these HITs are typically "renewed" by their requesters by the continuous posting of new HITs within the same HITgroup. (A common reason for this behavior is for the HIT to appear in the first page of the "Most recently posted" list of HITgroups, as many workers pick the tasks to work on from this list [4].) Although it is still unclear what dynamics causes this behavior, the analysis by Barabási indicates that priority-based completion of tasks can lead to such power-law distributions [2].

To better characterize this power-law distribution of completion times, we used the maximum likelihood estimator for power-laws. To avoid biases, we also marked as "censored" the HITs that we detected to be "aborted before completion" and the HITs that were still running at the last crawling date of our dataset (which will not be given in detail in this article).

The MLE estimator indicated that the most likely exponent for the power-law distribution of the completion times of Mechanical Turk is $\alpha = -1.48$. This exponent is very close to the value predicted theoretically for the queuing model of Cobham [3], in which each task upon arrival is assigned to a queue with different priority. Barabási [2] indicates that the Cobham model can be a good explanation of the power-law distribution of completion times only when the arrival rate is equal to the completion rate of tasks. Our earlier results indicate that for the AMT marketplace this is not far from reality. Hence the Cobham model of priority-based execution of tasks can explain the power-law distribution of completion times.

Unfortunately, a system with a power-law distribution of completion times is rather undesirable. Given the infinite variance of power-law distributions, it is inherently difficult to predict the necessary time required to complete a task. Although we can predict that for many tasks the completion time will be short, there is a high probability that the posted task will need a significant amount of time to finish. This can happen when a small task is not executed quickly, and therefore is not available in any of the two preferred queues from which workers pick tasks to work on. The probability of a "forgotten" task increases if the task is not discoverable through any of the other sorting methods as well.

This result indicates that it is necessary for the marketplace of AMT to be equipped with better ways for workers to pick tasks. If workers can pick tasks to work on in a slightly more "randomized" fashion, it will be possible to change the behavior of the system and eliminate the "heavy tailed" distribution of completion times. This can lead to a higher predictability of completion times, which is a desirable characteristic for requesters. Especially new requesters, without the necessary experience for making their tasks visible, would find such a char-

Figure 4: The posting (left) and completion rate (right) on AMT as a function of the day of the week.

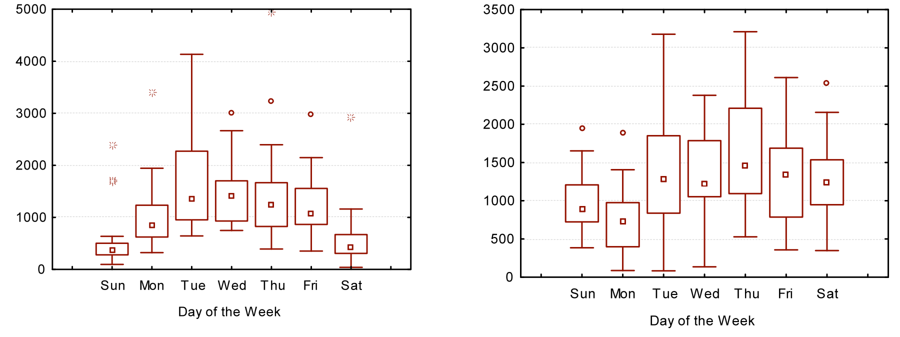
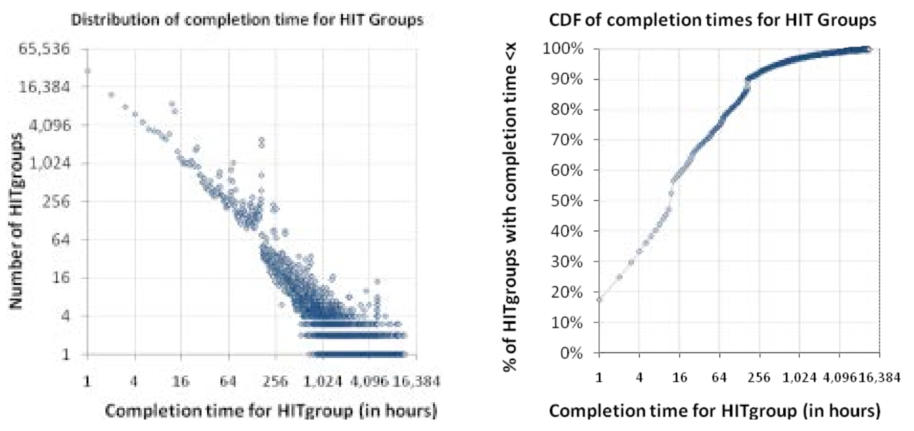


Figure 5: The distribution of completion times for HITgroups posted on AMT. The distribution does not change significantly if we use the completion time per HIT (and not per HITgroup), as 80 percent of the HIT groups contain just one HIT.



acteristic desirable, as it will lower the barrier to successfully complete tasks as a new requester on the AMT market.

We should note, of course, that these results do not take into consideration the effect of various factors. For example, an established requester is expected to have its tasks completed faster than a new requester that has not established connections with the worker community. A task with a higher price will be picked up faster than an identical task with lower price. An image recognition task is typically easier than a content generation task, hence more workers will be available to work on it and finish it faster. These are interesting directions for future research, as they can show the effect of various factors when designing and posting tasks. This can lead to a better understanding of the crowdsourcing process and a better prediction of completion times when crowdsourcing various tasks.

Higher predictability means lower risk for new participants. Lower risk means higher participation and higher satisfaction both for requesters and for workers.

WHAT DO WE KNOW ABOUT TURKER WORKERS?

Our analysis indicates that the AMT is a heavy-tailed market, in terms of requester activity, with the activity of the requesters following a log-normal distribution. The top 0.1 percent of requesters amount for 30 percent of the dollar activity and with 1 percent of the requesters posting more than 50 percent of the dollar-weighted tasks.

A similar activity pattern also appears from the side of workers [6]. This can be interpreted both positively and negatively. The negative aspect is that the adoption of crowdsourcing solutions is still minimal, as only a small number of participants actively use crowdsourcing for large-scale tasks. On the other hand, the long tail of requesters indicates a significant interest for such solutions. By observing the practices of the successful requesters, we can learn more about what makes crowdsourcing successful, and increase the demand from the smaller requesters.

We also observe that the activity is still concentrated around small tasks, with 90 percent of the posted HITs giving a reward of \$0.10 or less. A next step in this analysis is to separate the price distributions by type of task and identify the "usual" pricing points for different types of tasks. This can provide guidance to new requesters that do not know whether they are pricing their tasks correctly.

Finally, we presented a first analysis of the dynamics of the AMT marketplace. By analyzing the speed of posting and completion of the posted HITs, we can see that Mechanical Turk is a price-effective task completion marketplace, as the estimated hourly wage is approximately \$5.

Further analysis will allow us to get a better insight of "how things get done" on the AMT market, identifying elements that can be improved and lead to a better design for the marketplace. For example, by analyzing the waiting time for the posted tasks, we get significant evidence that workers are limited by the current user interface and complete tasks by picking the HITs available through one of the existing sorting criteria. This limitation leads to a high degree of unpredictability in completion times, a significant shortcoming for requesters that want high degree of reliability. A better search and discovery interface (or perhaps a better task recommendation service, a specialty of Amazon.com, can lead to improvements in the efficiency and predictability of the marketplace.

Further research is also necessary in better predicting how changes in the design and parameters of a task can affect quality and completion speed. Ideally, we should have a framework that automatically optimizes all the aspects of task design. Database systems hide all the underlying complexity of data management, using query optimizers to pick the appropriate execution plans. Google Predict hides the complexity of predictive modeling by offering an auto-optimizing framework for classification. Crowdsourcing can benefit significantly by the development of similar framework that provide similar abstractions and automatic task optimizations.

Biography

Panagiotis G. Ipeirotis is an associate professor at the Department of Information, Operations, and Management Sciences at Leonard N. Stern School of Business of New York University. His recent research interests focus on crowdsourcing. He received his PhD in computer science from Columbia University in 2004, with distinction, and has received two Microsoft Live Labs Awards, two best paper awards (IEEE ICDE 2005, ACM SIGMOD 2006), two runner up awards for best paper (JCDL 2002, ACM KDD 2008), and a CAREER award from the National Science Foundation. This work was supported by the National Science Foundation under Grant No. IIS-0643846.

References

1. Mechanical Turk Monitor, http://www.mturk-tracker.com.
2. Barabási, A.-L. 2005. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207-211.
3. Cobham, A. 1954. Priority assignment in waiting line problems. *J. Oper. Res. Sec. Am.* 2, 70–76.
4. Chilton, L. B., Horton, J. J., Miller, R. C., and Azenkot, S. 2010. Task search in a human computation market. In *Proceedings of the ACM SIGKDD Workshop on Human Computation* (Washington DC, July 25, 2010). HCOMP '10. ACM, New York, NY, 1-9.
5. Ipeirotis, P. 2010. Demographics of Mechanical Turk. working paper CeDER-10-01, New York University, Stern School of Business. Available at http://hdl.handle.net/2451/29585.
6. Ross, J., Irani, L., Silberman, M. S., Zaldivar, A., and Tomlinson, B. 2010. Who are the crowdworkers?: shifting demographics in mechanical turk. In *Proceedings of the 28th of the international Conference Extended Abstracts on Human Factors in Computing Systems* (Atlanta, Georgia, USA, April 10 - 15, 2010). CHI EA '10. ACM, New York, NY, 2863-2872.
7. M/M/1 model, http://en.wikipedia.org/wiki/M/M/1_model.

Crowdsourcing, Collaboration and Creativity

While many organizations turn to human computation labor markets for jobs with black-or-white solutions, there is vast potential in asking these workers for original thought and innovation.

By Aniket Kittur

DOI: 10.1145/1869086.1869096

Crowdsourcing has become a powerful mechanism for accomplishing work online. Hundreds of thousands of volunteers have completed tasks including classifying craters on planetary surfaces (http://clickworkers.arc.nasa.gov), deciphering scanned text (ReCaptcha http://recaptcha.net), and discovering new galaxies (http://galaxyzoo.org). Crowdsourcing has succeeded as a commercial strategy as well, with examples ranging from crowdsourcing t-shirt designs (Threadless http://www.threadless.com) to research and development (Innocentive http://www2.innocentive.com).

One of the most interesting developments is the creation of general-purpose markets for crowdsourcing diverse tasks. For example, in Amazon's Mechanical Turk, tasks range from labeling images with keywords to judging the relevance of search results to transcribing podcasts. Such micro-task markets typically involve small tasks (on the order of minutes or seconds) that users self-select and complete for monetary gain. These markets represent the potential for accomplishing work for a fraction of the time and money required by more traditional methods.

Crowdsourcing has worked especially well for certain kinds of tasks, typically, ones that are fast to complete, incur low cognitive load, have low barriers to entry, are objective and verifiable, require little expertise, and can be broken up into independent subtasks. Examples in Mechanical Turk include tagging images, evaluating search results, finding contact information, and labeling data for use in machine learning.

However, these simple tasks are only tappint the lowest levels of human intelligence. Are there opportunities to extend general purpose crowdsourcing markets to complex tasks that require significant intelligence or creativity? Innocentive, for example, has already shown that some companies may be willing to pay tens of thousands of dollars for one-off creative research and development—however, this represents a relatively small pool of highly expert workers and highly motivated companies.

The potential opportunity is enormous: tapping into hundreds of thousands or millions of workers across the globe to accomplish creative work on an unprecedented scale. In this article, I discuss ways to accomplish tasks that go beyond those that have traditionally been successful in micro-task markets, using subjective user studies and collaborative translation as examples. I also highlight some challenges and directions for future work that may be helpful for guiding other researchers interested in the field. I focus primarily on research-relevant tasks, given that those will be what most readers will find of most interest; however, many of the techniques presented here may apply to accomplishing complex work tasks as well.

CROWDSOURCING SUBJECTIVE TASKS

There are many potential tasks for which there is no one right answer. Researchers collect user ratings, elicit opinions, give surveys, and run experiments in which each individual's opinion may be different but equally valid. The possibilities of accomplishing tasks that don't have a predefined answer or solution through crowdsourcing in a matter of hours instead of weeks and for pennies instead of dollars is, with good reason, quite enticing. However, getting good results is no trivial matter. Subjective tasks

Illustration by Paul Wearing

are especially prone to users "gaming" the system, providing low-effort or random answers in order to reduce the time and work needed to get a reward. How would you know whether a given answer is a user's honest opinion or whether she is just clicking randomly?

My research partners and I first encountered this in 2007 when we tried to get turkers to rate the quality of Wikipedia articles, which has been a difficult thing for researchers to do with automated tools. With high hopes, we posted a number of articles to Mechanical Turk, paying turkers $0.05 to make judgments about their quality and to write what improvements they thought the article needed. We got results remarkably quickly: 200 ratings within two days.

But when we looked at the results, it was obvious that many people were not even doing the task. Almost half the responses to the writing question were blank, uninformative, or just copied and pasted, and about a third spent less than a minute on the task—less time than is needed to read the article, let alone rate it. This was not a promising beginning.

Is there something we could do to improve the quality of responses and reduce gaming? In traditional settings, there are a number of mechanisms that can mitigate gaming. For example, experimenters have long known that being in the same room as a participant can lead to better performance by the participant, as she knows the experimenter is monitoring her. Social norms, sanctions, and a desire to avoid looking bad can promote higher quality contributions in groups that know each other. The potential for additional future work can motivate high quality work in the present, as can formal or informal reputation systems that enable future employers to observe past performance. Job loss can create hardships, with time needed to find another job. Explicit contracts can be enforced through legal systems, causing high costs to those who do not honor them.

However, these mechanisms are less effective or absent in online micro-task markets such as Mechanical Turk. Social norms and sanctions are largely absent as workers are interchangeable and usually identified only by their worker ID—a random string of numbers and letters.

Interestingly, there are social sanctions of employers by the workers in Mechanical Turk, who use forums such as turkernation.com to alert other workers about bad employers (for more, see "Ethics and Tactics of Professional Crowdwork," page 39). This is partially in response to the asymmetrical power balance in Mechanical Turk: employers can reject work for any reason with no recourse for the worker whose has already put in the effort but does not get paid. Unification sites like Turkernation let workers sanction employers who consistently do not approve legitimate work.

New identities are relatively easily created, though they do require some external validity, such as credit card verification. Monitoring is essentially non-existent since the worker may be in any physical location in the world. Furthermore, employers don't know whether a worker who has accepted the task is actually engaged in it or is multitasking or watching TV.

Workers can choose between employers and jobs easily with no switching costs. Although some rudimentary reputation systems do exist (such as tracking the proportion of work rejected), workers with even low reputations can often find jobs to complete. There are no explicit contracts. Even after a worker has accepted a job, she can return it any time for any reason without consequence.

In the absence of external mechanisms for enforcing quality responses in subjective tasks, we turned to the design of the task itself. Specifically, we had two key criteria for task design. First, we wanted it to take the same amount of effort for a worker to enter an invalid but believable response as a valid one written in good faith. Second, we wanted to signal to the workers that their output would be monitored and evaluated.

To meet these criteria, we altered the rating task. Instead of subjective ratings followed by subjective feedback about what could be improved, we required turkers to complete three simple questions that had verifiable, quantitative answers, such as how many references/images/sections the article had. We also asked turkers to provide between four and six keywords summarizing the article. Importantly, we selected these questions to align with what Wikipedia experts claimed they used when rating articles (such as examining the references or the article structure), with the goal that by answering these questions, they would have a reasonable judgment of the quality of the article. We placed the verifiable questions before the subjective questions so workers would have the opportunity to develop this judgment before even having to think about subjective questions. Finally, since these questions have concretely verifiable answers, they signal that workers' responses can and will be evaluated—preventing gaming in the first place and potentially increasing effort (criteria 2).

"Instead of subjective ratings followed by subjective feedback about what could be improved, we required turkers to complete three simple questions that had verifiable, quantitative answers."

Re-running our experiment with the new task design led to dramatically better results. The percent of invalid comments dropped from 49 percent to 3 percent, improving by more than a factor of 10. Time spent on the task also more than doubled, suggesting increased effort. This was borne out by a positive and statistically significant correlation between turker ratings and those of expert Wikipedians. Finally, we found that we tapped a more diverse group, with more users contributing and a more even spread of contributions across users. (Details of the study can be found in "Crowdsourcing User Studies With Mechanical Turk," in *Proceedings of the ACM Conference on Human-factors in Computing Systems*, 2008.)

Although our initial results were rocky, in the end we found that crowdsourcing markets like Mechanical Turk can be quite useful for subjective tasks, as long as the task is designed appropriately to prevent gaming and ensure quality responses.

This opens up possibilities for entire new domains of tasks to be accomplished an order of magnitude faster and cheaper through crowdsourcing than traditional means. Imagine being able to do user testing of design prototypes and getting feedback within days from hundreds of users for dozens of prototypes, or running dozens of experiments a month, and accessing participants from across the globe rather than the low-diversity undergraduate participant pool available at most universities.

COLLABORATIVE CROWDSOURCING

One common assumption about Mechanical Turk is that turkers must work independently of each other. Most tasks involve turkers each making an independent judgment about an object (such as providing a label for an image) with their judgments aggregated afterward.

However, even interdependent tasks do not involve turkers interacting with each other. For example, the company CastingWords accomplished podcast transcriptions in a serial fashion: one turker may do the initial transcription; the transcription is automatically split into segments; other turker workers verify or improve the segments. Throughout, turkers never have to interact with each other despite using the results of each others' work. This is a reasonable approach when a requester does not know who will accept a task, when they will complete it, what the quality of the work will be, and when there are few dependencies such that work can be easily split up and done in parallel.

Figure 1: Workers collaborated to translate a poem using Etherpad, and the results are shown. Each color indicates a different worker's contribution.

```
POEM TO TRANSLATE
========================================

La luna vino a la fragua
con su polisón de nardos.
El niño la mira, mira.
El niño la está mirando.

En el aire conmovido
mueve la luna sus brazos
y enseña, lúbrica y pura,
sus senos de duro estaño.

TRANSLATE OR IMPROVE TRANSLATION BELOW
========================================

The moon came to the forge
with her bustle of flowering lilies.
The boy watches her, watches.
The boy is watchi her.

In the quivering air
the moon moves her arms
and reveals, lustful and pure,
her breasts of hard tin.
```

Many collaborative tasks in the real world, however, involve people interacting with each other. Examples range from scientists collaborating on a discovery to students collaborating on a report to volunteers writing articles together on Wikipedia.

In real-world collaborations, interaction is the norm rather than the exception. There are many advantages to interacting groups, such as the ability to communicate and coordinate on the fly rather than having to follow prespecified plans or rules, motivational gains from identifying with a group, and the bonds formed from interacting with other group members and helping behavior between them. There may be interesting benefits from breaking the assumption of independence and enabling workers to collaborate interactively in crowdsourcing.

However, such benefits are by no means certain. For example, would workers participating in a financial market really help each other without any financial incentives?

We examined this question in the context of a problem that is both difficult overall and especially difficult to do in a crowdsourcing context: collaborative translation. Unlike the short, simple, objective, and verifiable tasks that are typical of Mechanical Turk, translation can be complex, challenging, time-consuming, highly subjective, and impossible to verify automatically. It is also highly interdependent, requiring a consistent voice and approach throughout. However, if we were able to harness the power of the crowd for translation, there could be many potential benefits ranging from supporting disaster relief efforts (as already demonstrated by CrowdFlower and Samasource in crowdsourcing translation in the Haitian relief efforts, see page 10) to providing essential training data to help machine translation research.

In order to support turkers working interactively on translations we used Etherpad, an open-source platform for real-time collaborative editing that gave us the ability to track the participation patterns of each worker, support keystroke-by-keystroke real-time editing, show the contributions of each worker in a different color, and support chat between workers. Using this platform we had turkers come and add to or improve a translation of the famous Spanish poet Garcia Lorcas' poem "Romance de la luna," paying them $0.15 for their contributions. We started turkers with the original untranslated Spanish, as well as the first two sentences from a published English translation (see **Figure 1**).

As an aside, we found a marked improvement in acceptance and completion of the task when it was already "seeded" with a couple sentences that workers thought were contributed by others. This may be because it didn't seem quite as overwhelming, and new workers had a place to start; or it may have signaled that it was legitimate work; or that it was a reasonable amount of work since others had done it. Such seeding may be a useful technique for other tasks as well.

Within hours, more than a dozen turkers were working on the translation interactively, seeing each others' edits reflected in real time on the pad. After 48 hours, we took the results and had a different set of turkers compare it to a published translation, rating which they thought was a better translation of the original poem. Interestingly, 14 out of 16 bilingual raters pre-

Figure 2: Comparing the original poem (left), a published translation by Havard, 1990 (center), and final crowdsourced version from our experiment (right), the crowdsourced version was preferred by 14 out of 16 bilingual raters.

Romance de la luna, luna	Translation 1	Translation 2
La luna vino a la fragua con su polisón de nardos. El niño la mira, mira. El niño la está mirando.	The moon came to the forge in her bustle of spikenard. The boy stares at her. The boy is staring hard.	The moon came into the forge with her bustle of flowering lilies. The boy watches her, watches. The boy gazes upon her.
En el aire conmovido mueve la luna sus brazos y enseña, lúbrica y pura, sus senos de duro estaño.	In the feverish air the moon sways her arms, showing, lewd and spotless, her cruel, tin breasts.	In the quivering air the moon rotates her arms and, lustful and pure, reveals her breasts of hard tin.
Huye luna, luna, luna. Si vinieran los gitanos, harían con tu corazón collares y anillos blancos.	Run away, moon, moon, moon. If the gypsies find us, they would cut out your heart to make necklaces, silvery rings,	Fly moon, moon, moon, If the gypsies came, of your heart they would make white rings and necklaces.
Niño, déjame que baile. Cuando vengan los gitanos, te encontrarán sobre el yunque con los ojillos cerrados.	Child, let me dance. When the gypsies come, they will find you on the anvil with your tiny eyes shut tight.	Child, let me dance. When come the gypsies, on the anvil they'll find you with your lovely eyes closed.
Huye luna, luna, luna, que ya siento sus caballos.	Run away, moon, moon, moon. I can hear their horses.	Fly moon, moon, moon For I already feel their horses.

Figure 3: Amazon Mechanical Turk workers show an ability to communicate and coordinate surprisingly well on a translation project.

F.: Yeah... the elemnt is called "tin". 16:59
F.: Pewter refers to the metal alloy. 16:59

F.: "Lúbrica" translates as "lustful", not "playful" which would be "lúdica". 16:56

EMC: "nard"? I doubt many E 18:08
EMC: nglish speakers know what a nard is. the tuberose, or nardo, is in the family Liliáceas, so I translate it as lily, which is similar and more well known to English speakers 18:09

nefarious: OK, I made some different choices which I wiill explain here. 19:23
nefarious: The polisón can also be a crinoline, a hoop skirt. I imagine a wide skirt made of nard. 19:23
nefarious: Nard is different from lily, but not well known among English speakers. I had to look it up. 19:24
nefarious: It's in the valerian family, so with a little poetic license I called it Valerian, since it sounds nicer. 19:24
nefarious: Its main characteristic is an intense musky fragrance, so this is surely part of the poet's idea. 19:24
nefarious: Estaño can mean either tin or pewter, I imagine the moon's breasts more like pewter, sounds more poetic. 19:25

ferred the crowdsourced work over the published translation (see **Figure 2**). The total cost for this translation was less than $5.

There were many additional interesting things we found about the process. Certain passages were more difficult to translate than others, and we could see which these were as they were continually changed. For example, "En el aire conmovido" went through ten different revisions starting with "amidst the shaken air" and ending with "in the quivering air" by ten different contributors. In contrast, the first sentence ("La luna vino a la fragua") was translated as, "The moon came to the forge," and was not subsequently changed, suggesting it was easier to translate in the context of the poem. Such meta-information could provide valuable feedback for translation research or judging the ease or quality of translations.

The process of turkers working together interactively was at least as interesting. Many of them coordinated their edits with others, explaining and asking advice using chat. Some examples are shown in **Figure 3**. They also found the process enjoyable and rewarding, making statements such as "this was fun" or "sweet" in the chat, and emailing us further changes to the poem after the experiment was over, despite already having been paid.

Another unexpected outcome happened when we left the pad open after an experiment completed. Some turkers removed the original poem and replaced it with a new poem which others then came and translated for free! Together, these results show the potential for gains in effort, motivation, coordination, and quality that can be achieved by letting people work together collaboratively rather than treating them only as simple processors or algorithms. It's especially remarkable given the context of the environment, which was a market-based crowdsourcing platform in which the primary driver is money. Is there more value than we at first recognized in supporting the social nature of crowdsourcing, especially for creative tasks?

CROWDSOURCING RESEARCH TO COME

Crowdsourcing is already a powerful approach to solving a variety of problems faster and cheaper than traditional methods, but how far we can push it? For example, instead of treating crowd workers as simple sensors or processors, can we harness their human intelligence and creativity? Instead of focusing on simple, independent tasks, can we generalize to more complex tasks where people need to work together? In the limit, one can imagine a future in which the *de facto* method of doing work is through crowdsourcing markets, with experts and non-experts collaborating in *ad-hoc* groups to accomplish tasks.

We need to learn much more about the possibilities and limits of crowdsourcing. One important area of research may be understanding the motivational and reward structure of crowd workers and how they generalize across different kinds of markets. For example, workers on Mechanical Turk working for pennies are very different from those in Innocentive, the R&D crowdsourcing market in which solutions may pay tens of thousands of dollars.

Even for less extreme cases, such as people working for the kinds of virtual cash provided by Facebook games, such as Farmville, there are different demographics and motivations than those on Mechanical Turk. Furthermore, motivation may not be solely based on external rewards. Intrinsic motivations such as fun, camaraderie, and meaning may be just as powerful mechanisms for motivation, and may have especially beneficial effects on quality. However, more research is needed to understand the factors involved in task acceptance and completion.

Another important area of research is how to structure tasks to meet the needs of different markets. In Mechanical Turk, for instance, work that takes a long time, incurs high cognitive demands, or has an uncertain payment structure or description tends to be less attractive than short, simple tasks with high certainty of reward, even if the reward is lower overall. Parceling work into short, simple subtasks may be optimal in such markets, while chunking work to avoid the switching costs of people choosing another task to do may be better suited to other markets. Matching the structure of the work to the characteristics of the market could lead to faster throughput and better quality.

Crowdsourcing markets like Mechanical Turk already show enormous potential, but they could go much farther. There are plenty of opportunities for new researchers to help realize these possibilities in fields such as economics, sociology, psychology, computer science, human-computer interaction, and policy. And there has never been a better time to step out of the crowd and get involved.

Biography

Aniket Kittur is an assistant professor in the Human-Computer Interaction Institute at Carnegie Mellon University. He received his PhD in cognitive psychology from University of California-Los Angeles, studying the cognitive processes underlying sensemaking activities such as learning, memory, and insight. His research focuses on harnessing the efforts of many individuals to make sense of information together in ways that exceed individual cognitive capacities in domains including Wikipedia, crowdsourcing markets, and scientific collaboration. His research employs multiple complementary techniques, including experiments, statistical and computational modeling, visualization, data mining, and machine learning.

Heads in the Cloud

A professor and several PhD students at MIT examine the challenges and opportunities in human computation.

By Robert C. Miller, Greg Little, Michael Bernstein, Jeffrey P. Bigham, Lydia B. Chilton, Max Goldman, John J. Horton, and Rajeev Nayak

DOI: 10.1145/1869086.1869095

C*rowd computing* is quickly becoming an essential part of the technology landscape. Crowd computing encompasses the interaction among large numbers of people facilitated by software systems and networking technology. Crowds—and by "crowds," we literally mean a mass of people—are themselves the power that fuels sites like Wikipedia, Twitter, Intrade, and even online labor markets like Amazon Mechanical Turk.

One way to think about crowd computing is as the human analogue to cloud computing. Where the cloud provides access to elastic, highly available computation, and storage resources in the network, the crowd represents access to elastic, highly-available *human* resources, such as human perception and intelligence. Crowd computing offers the strength of software with the intelligence and common sense of human beings.

HUMAN COMPUTATION

One variant of crowd computing is human computation, which we define as using software to orchestrate a process of small contributions from a crowd to solve a problem that can't be solved by software alone.

Human computation was first popularized by Games With a Purpose (http://gwap.com), in which the computation is a side effect of a fun game **[8]**. For example, the *ESP Game* asks two players to guess words associated with an image, scoring points when their words agree, which makes the game fun, but also generating useful labels to index the image for searching, which makes it an act of human computation.

Another human computation site is Amazon Mechanical Turk (http://mturk.com), a marketplace where people get paid to perform human computation. Users, or "workers," find short tasks that are posted by "requesters" (the people who need the tasks completed) and get paid small amounts of money for completing them. CrowdFlower (http://crowdflower.com) is another site that pays users for computation—in not only real currency, but also virtual currencies for games like *Farmville* and *Mafia Wars*. Social networks like Facebook are also becoming platforms for human computation, motivated by social relationships rather than entertainment or monetary reward.

These platforms make it increasingly feasible to build and deploy systems that use human intelligence as an integral component. But there are at least three challenges to exploring the space of human computation systems: 1) applications—understanding what's appropriate for human computation and what isn't; 2) programming—learning how to write software that uses human computation; and 3) systems—learning how to get good performance out of a system with humans in the loop.

APPLICATIONS

What application areas will benefit the most from human computation? What properties do certain problems possess that make them amenable to a successful solution by a hybrid human-software system? Since the end user of such a system is also, typically, human, we can refine this question further: Why does a human end user need to request the help of a human crowd to accomplish a goal, rather than just doing it herself?

One reason is differences in capability. A group of many people has abilities and knowledge that one single end user does not, either innately or because of situational constraints. For example, VizWiz **[1]** helps blind users answer questions they have about things around them that they cannot see. The blind person takes a photograph with a smartphone's camera, records a spoken question (also using the phone), and then uploads the query and picture to a crowd of sighted users on the net who are better able to answer it (see **Figure 1**). For example, if a blind person grabs a can out of her cupboard but has forgotten what's inside it, she can snap a photo of the can and its label, upload it, and ask the sighted users what's in the can.

A related system, Sinch [7], draws on the crowd to provide assistance to web-enabled mobile device users who have situational disabilities, such as the limited ability to read a small screen, arthritis or hand tremors that make it difficult to click on small web page targets, and slow networks. With Sinch, the mobile users speak a question into their phone and the crowd searches the web for answers, using their more capable desktop web access, and returning web pages with the requested information highlighted.

Another reason to use a crowd is the "many eyes" principle, which has been claimed as an advantage of open-source software development (the complete phrase is "many eyes make bugs shallow"). We have exploited this principle in Soylent [2], a Microsoft Word extension that uses a crowd for proofreading, shortening, and repetitive editing. A typical run of Soylent may have dozens of people looking at each paragraph of a document, finding errors that a single writer might miss. In fact, a conference paper submitted about Soylent contained a grammatical error that was overlooked by not only Word's built-in grammar checker, but also eight authors and six reviewers. However, when we passed the paper through Soylent, the crowd caught the error.

A corollary of the many eyes principle is diversity. The fact is, a crowd comprises a wide range of ideas, opinions, and skills. For example in Soylent, the system not only indentifies writing errors, but also suggests multiple ways to fix them. It can suggest text to cut to save space—a tough task even for skilled authors, who are often reluctant to make cuts. Soylent can typically trim text down to 85 percent of its original length, without changing the meaning of the text or introducing errors (see **Figure 2**).

Figure 1: With VizWiz, blind people take photos using their mobile phones and submit them alongside a question, spoken orally into the phone, shown here above each image. A crowd of anonymous users reply, shown below, with response time given in seconds in parentheses.

What denomination is this bill?	Do you see picnic tables across the parking lot?	What temperature is my oven set to?	Can you please tell me what this can is?

(24s) 20 **(29s)** 20	**(13s)** no **(46s)** no	**(69s)** it looks like 425 degrees but the image is difficult to see. **(84s)** 400 **(122s)** 450	**(183s)** chickpeas. **(514s)** beans **(552s)** Goya Beans

PROGRAMMING

Prototyping a human computation system is hard if you have to entice a crowd to visit your website. Games With a Purpose handles this by making the experience fun—but not all human computation systems are fun enough to be self-motivating, particularly at the prototyping stage. Mechanical Turk is a good prototyping platform for many forms of human computation, because it offers a ready service for recruiting a crowd on demand. And the first prototypes for VizWiz and Soylent were built on Mechanical Turk.

Yet thinking about programming with human beings inside the system poses special problems. For example with Mechanical Turk, a request for a human to do a small task can take a few minutes and cost a few cents to get a result, which is astounding in one sense (that you can obtain human assistance so quickly and so cheaply), but is abysmally slow and expensive compared to a conventional function call.

Programmers need new tools that can help them experiment with human computation in their systems. For example, our TurKit toolkit [3] integrates Mechanical Turk calls in a traditional imperative/object-oriented programming paradigm, so that programmers can write algorithms that incorporate human computation in a familiar way. TurKit does this using a novel programming model called "crash and rerun," which is suited to long-running distributed processes where local computation (done by software) is cheap, and remote work (done by humans) is costly.

The insight of crash-and-rerun programming is that if our program crashes, it is cheap to rerun the entire program up to the place where it crashed. This is true as long as rerunning does not re-perform all the costly external operations from the previous run. The latter problem is solved by recording information in a database every time a costly operation is executed.

Costly operations are marked by a

"A group of many people has abilities and knowledge that one single end-user does not... The fact is, a crowd comprises a wide range of ideas, opinions, and skills."

Figure 2: In Soylent, after the crowd has suggested words or phrases that can be edited, the end-user can shorten his or her text interactively with a slider. Red text indicates locations where cuts or rewrites have occurred.

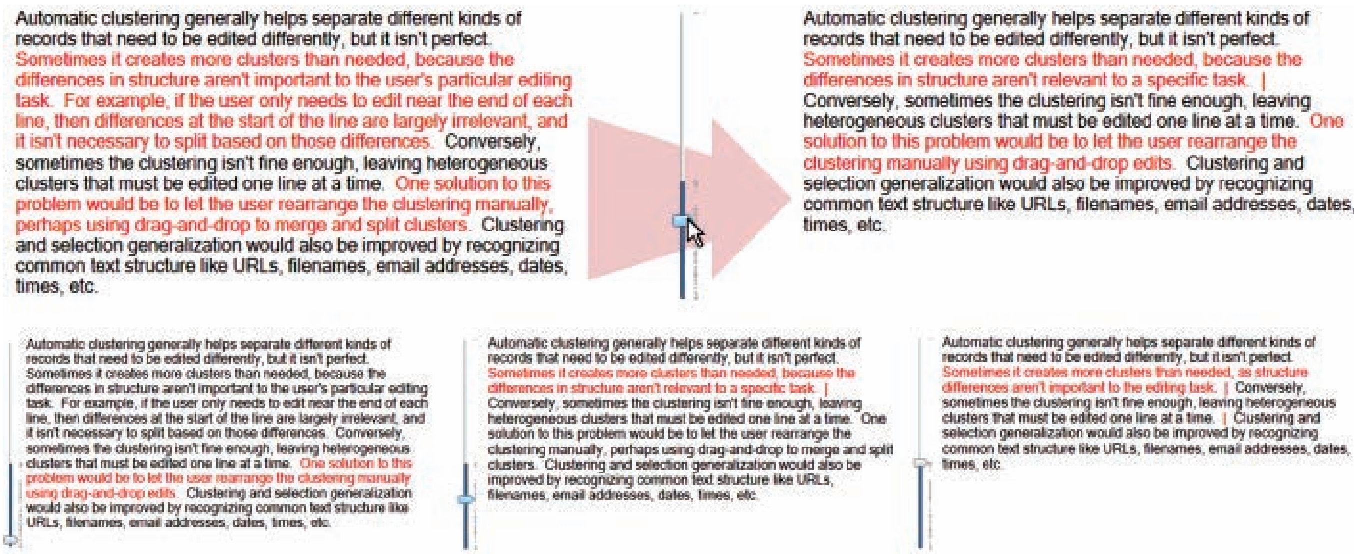
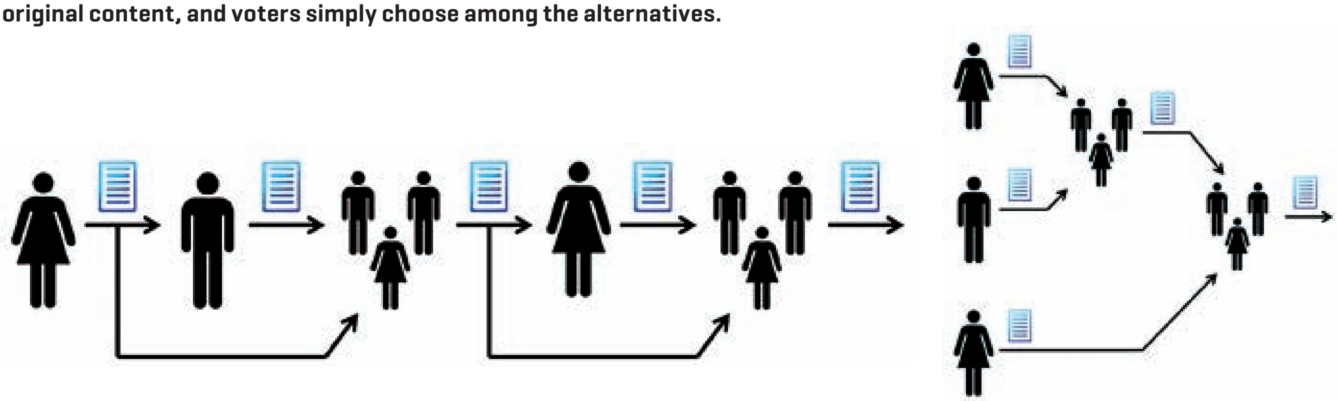


Figure 3: Some human computation processes are iterative (left), involving a succession of interleaved improvement steps (by one person) and voting steps (by several people). Other processes are parallel (right), in which individuals generate original content, and voters simply choose among the alternatives.



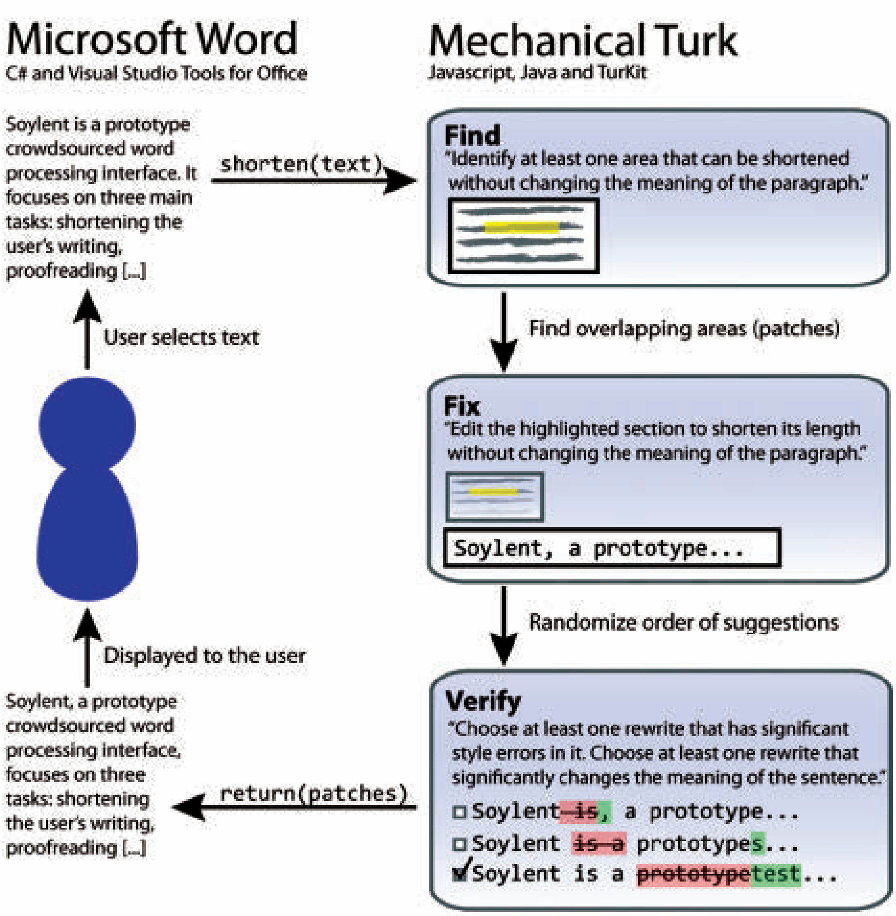
new primitive called "once," meaning they should only be executed once over all reruns of a program. Subsequent runs of the program check the database before performing operations marked with "once" to see if they have already been executed. This model makes it much easier to code algorithms involving human computation. For example, a TurKit program can sort a list of images using human preference judgments by calling the human computation in the sort algorithm's comparison function, and wrapping those calls in "once" to make them persistent.

Another programming challenge is the development of algorithms and design patterns that handle the idiosyncrasies of human beings. Humans are not programmable machines, and they don't always follow instructions, unintentionally or otherwise. Sometimes this should be embraced and supported, to harness the creativity and diversity of the crowd. Other times, it simply produces noisy, erroneous, or useless results.

For example, we have studied alternative algorithms for content creation [**4**]. Iterative processes are similar to Wikipedia or open-source software development. People build on existing content created by others, with voting or independent review ensuring that the process stays on track. Parallel processes are often seen in design contests, like Threadless.com, where people generate content independently, and then the best is chosen through a vote. See **Figure 3**.

In experiments involving various kinds of work, such as handwriting transcription, image description, and brainstorming, our results show that iterative processes generally produce higher than average quality than parallel processes. However, in the case of brainstorming, workers riff on good ideas that they see to create more good ones, but the very best ideas seem to come from people working alone. And

Figure 4: The find-fix-verify algorithm in Soylent identifies patches in need of editing, suggests fixes to the patches, and votes on those fixes.



with transcription tasks, it turns out that showing workers the guesses of other workers often leads them astray, especially if the guesses are self-consistent but wrong.

Crowd workers exhibit high variance in the amount of effort they invest in a task. Some are lazy turkers, who do as little work as necessary to get paid, while others are eager beavers, who go above and beyond the requirements, either to be helpful or to signal that they aren't lazy turkers, but in counterproductive ways. We need new design patterns for algorithms involving human computation that recognize and control this behavior.

For example, Soylent uses a find-fix-verify pattern to improve the quality of proofreading and document shortening (**Figure 4**). In this pattern, some workers find problems, other workers fix them, and still other workers verify the fixes. But questions remain. What other algorithms and design patterns are useful? How should algorithms involving human computation be evaluated and compared from a theoretical point of view?

SYSTEMS PROBLEMS

Moving from prototyping to actual deployment requires facing questions about how to obtain a reliable and well-performing source of human computation for the system. How can we recruit a crowd to help, and motivate it to continue to help over time, while optimizing for cost, latency, bandwidth, quality, churn, and other parameters?

For paid crowds, these questions intersect with labor economics. Some of our recent work has found that workers in human computation markets like Mechanical Turk behave in unusual ways. For example, instead of seeking work that provides a target wage, they often seek a target earning amount, and simply work until they reach their target, consistent with game-playing behavior [**5**].

Another difference in these markets is the overwhelming importance of searchability. Workers' ability to find tasks they want to do is strongly affected by the kind of interface the market offers. Mechanical Turk, for example, typically displays a list of thousands of available tasks, divided into hundreds of result pages, with few effective tools for searching or filtering this list. We have found that most workers simply choose a particular sort order and work their way through the list. They most often sort by newest task, or most tasks available, and surprisingly not by price. The speed of completion of a task is strongly affected by its ability to be found, which may not be strongly related to the monetary reward it offers [**6**].

We can also think about human computation in computer systems terms, such as cost, latency, and parallelism. Services like VizWiz and Sinch need to return answers quickly, and to support that, we have developed an approach (and accompanying implementation) called quikTurkit that provides a layer of abstraction on top of Mechanical Turk to intelligently recruit multiple workers before they're needed.

In a field deployment of VizWiz, users had to wait a little longer than two minutes on average to get their first answer. Wait times decreased sharply when questions and photos were easy for workers to understand. Answers were returned at an average cost per question of only \$0.07 for 3.3 answers. Given that other visual-assistance tools for the blind can cost upwards of \$1,000 (the equivalent of nearly 15,000 uses of VizWiz), we believe that human computation embedded in an inexpensive software system can be not only more effective but also competitive with, or even cheaper than, existing pure software solutions. When set to maintain a steady pool of workers (at a cost of less than \$5 per hour), quikTurkit can obtain answers in less than 30 seconds.

Beyond monetary compensation, many other reasons entice people to participate in human computation, including altruism, entertainment, and friendship. How do those motivations influence system performance? And how should the systems be designed to encourage some motivations, and perhaps discourage others?

After demonstrating that VizWiz was feasible using paid strangers on Mechanical Turk, we also ported it to Facebook, so that a blind user's sighted friends can help. We are currently studying how people (at least in this context) choose to trade off the strengths and weaknesses of each service. Mechanical Turk is fast but costs money. Facebook is free, and the user's friends might be more motivated to answer, or even more capable since they know more about the person. On the other hand, the user might be less willing to ask certain personal questions to his or her friends, rather than asking an anonymous Mechanical Turk crowd.

PEOPLE VS SYSTEMS

The gap between what software can do and what people can do is shrinking, but a gap of some sort will exist for a long time. Automatic techniques need to be able to fallback to people when necessary to fill in the gaps, enabling interactions that automatic techniques alone can't yet support and helping us design for the future.

So-called *Wizard of Oz* prototyping (wherein a human is hiding behind the curtain, so to speak) is a venerable technique in human-computer interaction and artificial intelligence that makes an intelligent system (or even a not-so-intelligent one) appear to work even though a software backend isn't ready yet. With platforms like Mechanical Turk and Facebook that make human computation practical, we are now at the point where *Wizard of Oz* is not just for prototyping anymore. We can build useful systems with human power inside, and actually deploy them to real users. These systems will stretch the limits of what software can do, and allow us to find out whether the ideas even work and how people would use them.

In addition, we can collect data from actual system use, like VizWiz queries and photos, that might eventually help to replace some or all of the human power with artificial intelligence. From this perspective, AI would speed up performance and reduce labor costs. But human computation made the system possible in the first place.

Acknowledgements

Ideas and work in this article come from many students and collaborators, including Mark Ackerman, David Crowell, Bjoern Hartmann, David Karger, Marc Grimson, and Katrina Panovich. This work was supported in part by Quanta Computer, NSF, and Xerox.

Biographies

Robert C. Miller is an associate professor of computer science at Massachusetts Institute of Technology. He grew up in rural Louisiana and plays the accordion, although extremely poorly.

Danny "Greg" Little is a PhD student in computer science at MIT. He is probably asleep right now.

Michael Bernstein is a PhD student in computer science at MIT and a features editor for this magazine.

Jeffrey P. Bigham is an assistant professor of computer science at University of Rochester, and he wants to be mayor of your living room. Don't let him in.

Lydia B. Chilton is a PhD student in computer science at University of Washington. When not studying human computation, she is a consultant to the United Federation of Planets and makes the occasional journey on the USS Enterprise with her old pals Kirk, Spock, and McCoy.

Max Goldman is a PhD student in computer science at MIT. When he gives a talk, his performance is so lively and engaging that it distracts the audience from the actual research results, but everybody goes away happy.

John J. Horton is a PhD student in public policy at Harvard University. He is having trouble thinking of a public policy angle for his human computation research and needs to defend his dissertation soon. This is a growing problem that, to date, Turkers have been unable to solve.

Rajeev Nayak is a graduate student in computer science at MIT. He bats .400, shoots .575 from the field, sings *a cappella*, and watches *Glee* religiously.

References

1. Bigham, J.P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R.C., Miller, R., Whyte, B., White, S., Yeh, T. VizWiz: Nearly Real-time Answers to Visual Questions. *UIST 2010*.
2. Bernstein, M., Little, G., Miller, R.C., Hartmann, B., Ackerman, M.S., Karger, D.R., Crowell, D., Panovich, K. Soylent: A Word Processor with a Crowd Inside. *UIST 2010*.
3. Little, G., Chilton, L., Goldman, M., Miller, R.C. TurKit: Human Computation Algorithms on Mechanical Turk. *UIST 2010*.
4. Little, G., Chilton, L., Goldman, M., Miller, R.C. Exploring Iterative and Parallel Human Computation Processes. *HCOMP 2010*, to appear.
5. Horton, J.J. and Chilton, L. The Labor Economics of Paid Crowdsourcing. *EC 2010*.
6. Chilton, L., Horton, J.J., Miller, R.C., Azenkot, S. Task Search in a Human Computation Market. *HCOMP 2010*, to appear.
7. Nayak, R., *et al.* Sinch: Searching Intelligently on a Mobile Device. *CHI 2011*, in submission.
8. von Ahn, L., and Dabbish, L. Designing Games with a Purpose. *CACM*, 51, 8, August 2008.

ACRONYMS

AMT Amazon Mechanical Turk: a web service owned by Amazon that facilitates crowdsourcing

CAPTCHA complete automated public Turing test to tell computers and humans apart; it's a contrived acronym intentionally redolent of the word "capture," used to describe a test issued on web forms to protect against automated responses

GWAP Game with a Purpose: a computer game that layers a recreational challenge on top of a problem that demands human intelligence for efficient solution, e.g., protein folding

HCIR human–computer information retrieval

HIT human intelligence task: a task that an AMT requester is willing to pay to have accomplished by AMT providers. More generally, a task that may be best completed via crowdsourcing

HuGS human-guided search: A research project investigating a strategy for search and optimization problems that incorporates human intuition and insight

reCAPTCHA a kind of "CAPTCHA" (see above) service that helps to digitize books, newspapers and old radio shows. Used to detect whether a user is a human or a computer (bot)

TF-IDF term frequency-inverse document frequency: A weight formula used in data mining to determine the importance of a particular term to a document in a corpus of text

Mathematics for the Masses

Can human computation bring together people from diverse backgrounds to solve age-old math problems?

By Jason Dyer

DOI: 10.1145/1869086.1869097

Becoming involved in research outside the realm of academics has been historically difficult. In the field of mathematics, this difficulty stems not only from a lack of physical resources, but also from a lack of community. Mathematicians rely on their colleagues much more than outsiders might realize. However, with the recent rise of Polymath projects, the barrier to entry into mathematical research has been significantly lowered.

The first Polymath project was started on the blog of mathematician Timothy Gowers, and its purpose was to find a new proof for the Density Hales-Jewett Theorem. The intent was "massively collaborative mathematics," inviting anyone at all to become involved if they so desired. As Dr. Gowers explains, "When you do research, you are more likely to succeed if you try out lots of ideas, even if it is often the case that soon afterwards you can see why they never had a chance of working."

The end result was an important new proof, noted by publications like *Nature* and *The New York Times*. The entirety of the effort, including not only brilliant insights but mistakes, immediate retractions, and dead ends, is still online.

Even "foolish" ideas are welcome during a Polymath project because they might lead to more coherent ideas. For example, at one point I made a conjecture that was clearly, outrageously wrong. But it led us to discover a bug in our computer program to generate data.

COLLABORATIVE MATH

Let's explore a mathematical problem that is conducive to community collaboration. Consider some apple and orange trees planted in a row. Imagine starting at tree no. 1 and walking past all the trees, keeping track of how many of each type you passed along the way.

Find the largest difference between the number of apple and orange trees at any point during the walk. For the example shown in **Figure 1**, the largest difference is 2.

> **"When you do research, you are more likely to succeed if you try out lots of ideas, even if it is often the case that soon afterwards you can see why they never had a chance of working."**

Now, instead of including every tree, we could include only trees nos. 2, 4, 6, and so on—trees where the position is a multiple of two. See **Figure 2**.

This time the largest difference is 1.

We could do the same thing with multiples of 3, 4, 5, or any natural number. If we consider all these methods of counting, the largest difference possible is called the **discrepancy**.

Puzzle #1: Design a row of 11 apple and orange trees so that the discrepancy is 1. That is, checking every tree or any multiple, the difference between the number and apple and orange trees is always 1 or 0.

Puzzle #2: Prove Puzzle #1 can't be solved with 12 trees.

According to the solution of Puzzle #2, if we want a discrepancy of 1, at some point we have to stop planting trees.

Well, what about a discrepancy of 2? That turns out to break at 1,124—*probably*. As of this writing, no one has proved that it is impossible to plant

1125 trees such that the discrepancy is less than 2.

How about a discrepancy of 3, 4, or 5? Now the problem is getting to the point where we aren't sure how many trees we could have, but the evidence suggests that eventually we'll have to stop planting trees for those, too.

No matter what discrepancy we pick, there will always be a limit to the number of trees in the row. This is known as the Erdős discrepancy problem, and has remained unsolved for nearly 80 years.

HARNESSING HUMANS

Polymath projects continue apace. An effort to break the Erdős discrepancy problem is one of a number of works in progress. Everyone is invited to post ideas, no matter how small or insignificant they might seem.

While expertise helps, amateurs have made strong contributions. The discussion can get technically daunting, but this can even happen to mathematicians as experts in one field try to communicate with experts in another. Often there is some "side thread" of research, perhaps of an experimental sort, simply playing with data via computers at a much more elementary level. The experimental gain has been tremendous and any computer scientists interested specifically in mathematics are warmly welcome.

The Polymath ethos recently expanded further when Vinay Deolalikar released a proof that P does not equal NP, possibly cracking one of the most infamous unsolved problems in computer science. Within a week, a flurry of online activity led to lengthy blog threads and a comprehensive guide to the solution (and alas, some serious and presently unresolved objections).

There are still many more mathematical projects conducive to an open format. Both mathematicians and computer scientists should start to consider if the power of the masses might help with their own research.

Figure 1: In this visual introduction to the Erdős discrepancy problem, this example shows the counting increment as 1 and the largest difference as 2.

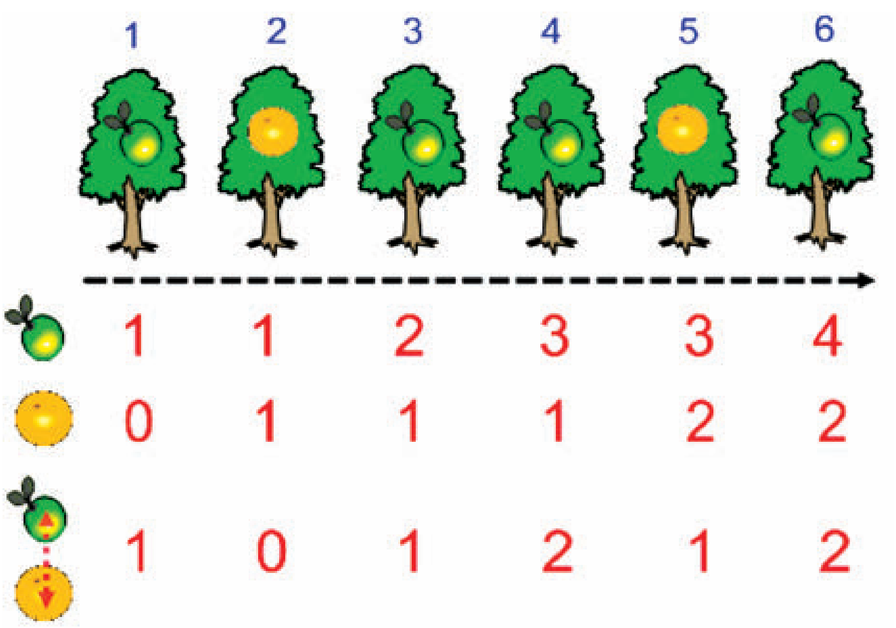
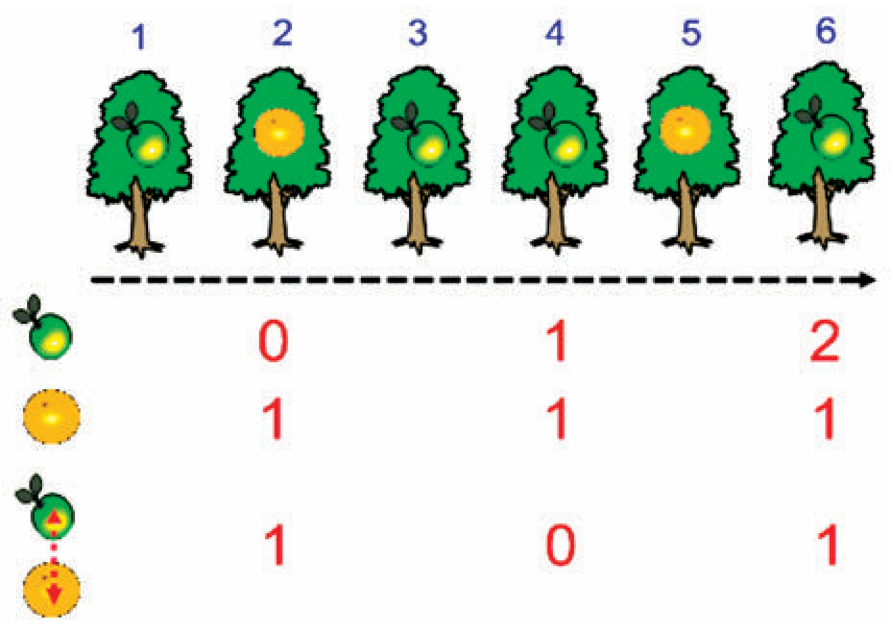


Figure 2: A second example the Erdős discrepancy problem explains discrepancies. In this example, the counting increment is 2, and the largest difference is 1.



Biography

Jason Dyer is an educational specialist at the University of Arizona with interests in the use of multimedia in the presentation of mathematics and the psychology of mathematical cognition. His past projects include Off-Road Algebra, a video series designed to teach algebra to middle school students via motocross racing. He writes at The Number Warrior at http://numberwarrior.wordpress.com.

Resources

1. Polymath blog: http://polymathprojects.org/.
2. Polymath wiki: http://michaelnielsen.org/polymath1/index.php?title=Main_Page.

An Introduction to Human-Guided Search

Can people help computers solve challenging optimization problems?

By Michael Mitzenmacher

DOI: 10.1145/1869086.1869098

Computers are often described as computational tools for humans. But what if we, in turn, can help the computers help us solve our problems? Many optimization problems are tackled in practice using some type of local search. The search algorithm has a current solution, and it tries to improve the solution by moving one of the problem elements to a neighboring solution.

For example, consider a system for optimizing deliveries with a fleet of trucks. The goal might be to reduce costs by minimizing the total number of miles driven. A solution could consist of the ordered list of deliveries for each truck, and the cost of a solution would be the total miles driven. In this case, moving a problem element could correspond to either moving a delivery from one truck route to another truck route, or changing the delivery order within a route. As we move from solution to solution, we aim to drive fewer total miles.

For several years I was involved in a project that considered the potential for interactive optimization systems, where humans assisted computers in finding the best solution. Our human-in-the-loop project was called HuGS, short for human-guided search.

Most people want to know what the motivation is for involving humans in these sorts of search problems. Why not just let the computer do the job?

MOTIVATING FACTORS

Several reasons for including humans are based on realistic real-world necessities. People often need to understand and trust a solution in order to implement it effectively. Being part of the solution process engenders that trust. Alternatively, the solution might need to be explained or justified to others in order to gain their cooperation. Having a person explicitly part of the process may improve explanations or help promote the result.

Another reason for including humans is it may be hard to accurately encapsulate full knowledge of the system being optimized into a computer program. Optimization systems typically solve an approximation or idealization of the actual problem of interest. People often know more about a problem than they can specify in advance or using a given problem-specification language. Allowing a human in the loop provides a mechanism for such external information to play a part in the solution.

While these "soft" motivations inspired us in this project, we also had a "hard" motivation. We hoped that putting humans in the picture would actually improve performance, in terms of achieving a better solution score in a shorter amount of time. It might seem that this could not be possible. Computers are blindingly fast at simple, repeated tasks, like trying to find moves to better solutions for an optimization problem—far faster than people.

However, there are still many tasks at which humans, for now, perform better than machines, including visual perception and strategic thinking. Our hypothesis was that, for many problems, allowing a person to steer a local search algorithm would improve performance. Essentially, we wanted to have the best of both worlds: humans provide big-picture thinking, while computers provide brute-force computational power.

The true goal was for a virtuous cycle, with humans giving repeated feedback to the computer, and vice versa, to home in on optimal or near-optimal solutions.

INTERFACES

A key question in putting humans in the process revolves around the user interface. Without a reasonable user interface, you cannot expect people to utilize the system effectively. We developed a toolkit that easily adapted to several problems and provided a user interface that combined simple metaphors and visualizations.

One key metaphor we developed was the use of mobilities, which provide a general mechanism to allow users to annotate solutions to guide the local search. Each element is assigned a mobility: high (green), medium (yellow), or low (red). The colors are meant to evoke a stoplight. Low-mobility elements are not allowed to move; they are essentially frozen. High-mobility elements can move in any way that does not move a low-mobility element. Medium-mobility elements can move only if made to do so by moving a high mobility element.

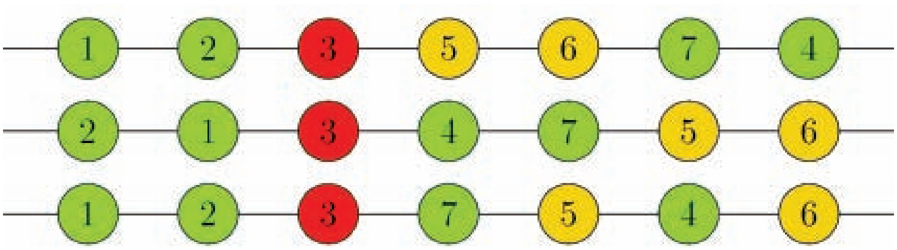
MOBILITY EXAMPLE

We can demonstrate mobilities with a simple example. Suppose the problem instance contains seven elements, and its solutions are all possible orderings of these elements. The only allowed move on an element is to swap it with an adjacent element. Suppose the current solution is as depicted in **Figure** 1, where we have assigned element 3 low mobility, elements 5 and 6 medium mobility, and the rest of the elements high mobility.

With these settings, our search algorithm can swap a pair of adjacent

Figure 1: This mobilities example shows the current solution. Red elements have low mobility, yellow equals medium mobility, and green equals high mobility.

Figure 2: This mobilities example shows several possible solutions that can be reached from the current solution (Figure 1) without violating mobility constraints.



elements only if at least one has high mobility and neither has low mobility. It is therefore limited to the space of solutions reachable by a series of such swaps, including those depicted in **Figure 2**.

Here, setting element 3 to low mobility essentially divides the problem into two smaller sub-problems. Also, while medium-mobility elements can change position, their relative order cannot be changed. Mobility constraints can drastically reduce the search space; for this example, there are only 12 possible solutions, while without mobilities, there are 7!=5,040 possible solutions.

Users could try to improve a solution by 1) manually moving elements and setting mobilities from a current solution, 2) invoking, monitoring, and halting a focused search for a better solution, or 3) reverting to previous solutions. As an example, in our original delivery application, a user might let a search algorithm run for some time, displaying the current best solution every few seconds. The user might then halt the search, and focus it by setting the mobility of several deliveries to low, freezing them on their current truck and in their current relative order. The user could then restart the search.

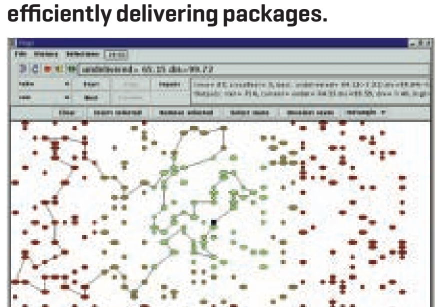
After a few seconds, if the improvement was minimal, the user might revert back to the last halting point, reposition some deliveries while freezing some others, and start again. The user might even switch the type of search being used when restarting, as our implementation provided several search methods as options.

We found that human-in-the-loop optimization worked as we hypothesized for several optimization methods and several applications. For example, one application we tested was a gathering problem (similar in spirit to the delivery problem), where a problem consists of a starting point, a maximum distance that can be traveled, and a set of customers each at a fixed geographic location with a given number of packages. The goal is to gather as many packages as possible without exceeding the maximum distance. In experiments with users, we found that on average it took over an hour of unguided search to match or beat the result from just ten minutes of search with human guidance. **Figure 3** shows an example of the display for the delivery application.

HOW FAR CAN WE GO?

In my mind, our project successfully demonstrated the power of putting humans in the loop for optimization problems where there is a natural visual representation. A little human time and effort can go a long way! But there are still many open questions. One of the most interesting is whether we can develop algorithms that learn from our interactions. When someone manually intervenes to help guide the search algorithm to a better solution, can the computer learn what strategic methods the human is using and adopt them for its own use?

Figure 3: The delivery application uses human-guided search to find routes for efficiently delivering packages.



Another question is whether human guidance can be valuable for problems with less natural visualizations, or for problems of a much bigger size where proper visualizations become more challenging. A related question is whether people can still have an impact as algorithms and machines become faster, expanding their comparative speedup over humans.

Even if humans cannot provide an actual performance gain, the issues related to humans needing to trust and implement the solutions the computer finds suggests that human-in-the-loop optimization remains a worthwhile challenge.

For a more technical survey of the HuGS project, see "Human-guided tabu search" [3]. Two good representative papers from the project, focusing on the performance results and the user interface [1, 2], are in the References.

Biography

Michael Mitzenmacher (http://www.eecs.harvard.edu/~michaelm/) is a professor of computer science in the School of Engineering and Applied Sciences at Harvard University. His work focuses on algorithms for networks and the Internet. His textbook on random algorithms and probabilistic analysis in computer science was published in 2005 by Cambridge University Press.

Acknowledgments

The author acknowledges the many people he had the pleasure of working with on HuGS, but especially Neal Lesh, Joe Marks, and Gunnar Klau. This work was supported by NSF grants CCF-0915922 and IIS-0964473.

References

1. Klau, G. W., Lesh, N., Marks, J., and Mitzenmacher, M. 2002. Human-guided tabu search. In *Proceedings of the 18th National Conference on Artificial Intelligence* (AAAI'02). AAAI Press. 41-47.
2. Klau, G. W., Lesh, N., Marks, J., Mitzenmacher, M., and Schafer, G. T. 2002. The HuGS platform: A toolkit for interactive optimization. In *Proceedings of Advanced Visual Interfaces*. 324-330.
3. Klau, G. W., Lesh, N., Marks, J., and Mitzenmacher, M. 2010. Human-guided tabu search. *Journal of Heuristics*, 16:3, 289-310.


BI-RITE

Beyond Freebird

Exploring Twitter and live events by structure and context can shed light on what people think.

By David A. Shamma

DOI: 10.1145/1869086.1869099

Four years ago, fresh out of graduate school, I started my first job as a research scientist at a large Internet company. People, social networks, media, and what was soon to be known as "big data" was on a slow rise to its boiling point. The Sunday before I started work was the final match of the 2006 World Cup: Italy versus France. Not even unpacked yet, I made my way to Dolores Park in San Francisco where I heard the game would be shown. Some 3,000 people were estimated to attend the event in the medium-sized park. I arrived to the crowd 10,000 strong.

It was so crowded that I could hardly make out Zidane's game-ending red card from my vantage point on top of a hill. After the game, I wondered why so many people would sardine themselves to watch but not really watch the game. What socially drives people to experience media together? Why is sharing conversation in context as important as watching the game itself? How does the conversation change how we perceive the event?

Fast forward to the 2010 World Cup. People gathered worldwide in bars, restaurants, homes, parks, and other places to watch the matches. Something new happened. In real time, people also shouted the cheers, the agony, and the "GOOOOOAAALLL"s across the internet via Flickr photos, social network updates, and microblog posts. Twitter, microblogging's principal representative, saw record-breaking post numbers just south of 3,000 tweets a second where their normal average is about 750. The mere fact that some 2,000 people a second spread around the world can converge around a single event broadcast from Johannesburg is incredible.

But, how do we start to understand what it means? How does one soccer match project meaning into a largely disconnected sea of voices? Trivially, one could suggest a spike in people saying "goal" means a goal occurred. This would likely work for simple event detection, like identifying a rock concert encore by the swell of people shouting, "Freebird," regardless of what band is playing.

Elections, concerts, and sporting matches all carry online conversations which, when viewed only from a Twitter stream, envelop the event and alter our perception. We should aim to understand both the conversation and the event. Is it possible to discern the structure of a media event from the conversation that is taking place around it?

With my Yahoo! Research colleagues Lyndon Kennedy and Elizabeth F. Churchill, we have begun to examine the relationship between the media event and the impact the event has on the structure and content of the concurrent but remote conversations that it engenders. This is the media's conversational shadow: the backchannel of social conversation cast from a shared event context. The goal is to attain a set of targeted measures through which we will be able to predict audience behaviors as events unfold. This should inform us how to design for participation—which should begin to blur the distinction between the event and its shadow. Before we can dive in, we have to understand the underlying structure and description of the conversation.

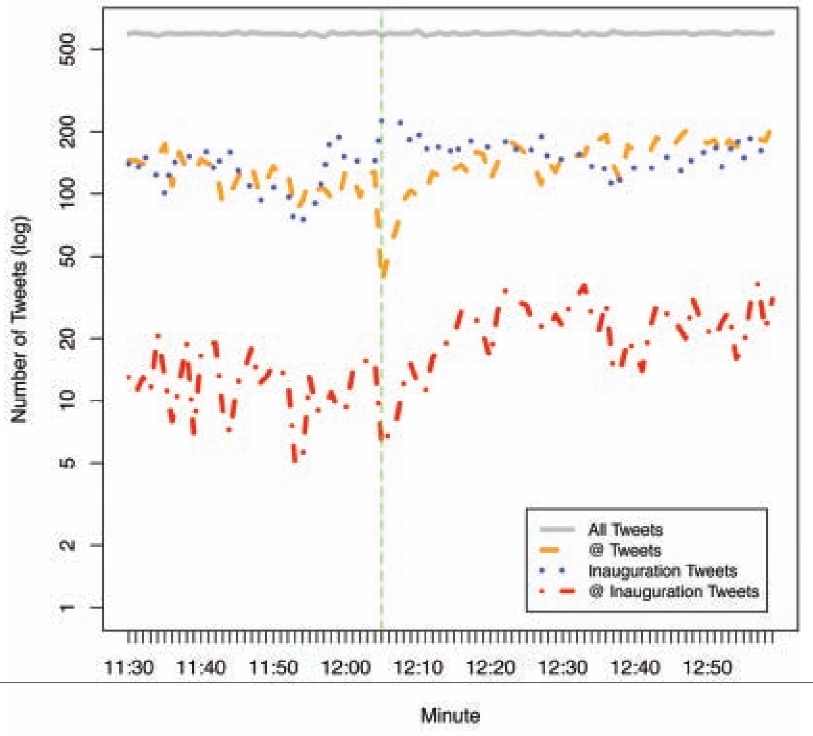
TALKING THROUGH THE MOVIE

The primary intuition, which is derived from human-centered social TV research [1], is simple. People stop with the conversation when something important happens—the hush at the start of a movie or that gasp at a missed goal. We have seen the same pattern in online chat rooms with shared video. Viewers tend not to communicate with each other while the video is playing, often saving conversations for logical breaks or lulls. Which is to say, the directed conversation comes to a stop. Presumably, when you do this, you have something to say which takes priority over the event.

A while back, I collected a very small handful of tweets from the first 2008 U.S. presidential debate. I wanted to see if people tweeted more *after* the debate, so I plotted the number of tweets per minute. There was a three-fold increase right when the debate stopped!

Even more interesting, the volume of tweets per minute during the debate was rather periodic and sinusoidal, which made me wonder if the contour hinted at the start and stop of each debate question segment. I used Newton's Method to find the roots of the graph. Then, using a seven-minute sliding window, I selected only the extrema greater than 1.5 standard deviations from the mean. The resulting set of minutes work like "cuts" as question

Figure 1: The fluctuations in volume of Twitter messages sent during U.S. President Barack Obama's inauguration show moments in time when users paid more attention to the ceremony (drop in volume) than to their computer use.



topic boundaries with about 92 percent accuracy [2]. All this is done without looking at any text.

The question then became, "Can this method of 'naïve tweet counting' scale?"

TWITTER'S EXPLOSION

Twitter has grown at an incredibly fast rate. In January of 2009, CNN Breaking News had around 86,000 followers. Nine months later, its follower count had exceeded 2.7 million. Twitter began offering data-mining push feeds that hands back 600 tweets a minute. Track feeds push all the tweets related to a search, and Twitter's "Firehose" can deliver everything if you can get your hands on it. I previously sampled 3,000 tweets from a 90-minute debate. I saw 52,000 tweets from a 90-minute sample of President Obama's inauguration speech.

Addressing scale, I revisited the human-centered observation I originally questioned at Dolores Park. If we position people tweeting while watching a live event, either in person or on television, conversation is represented as talking to someone else: in effect directing a tweet with the @ symbol (using "@" before a username like @barackobama pushes a highlighted tweet into that person's feed). The highlighted tweet becomes more visible; it calls attention to the user, like tapping your friend on the shoulder in a crowd.

Aggregating the number of @ symbols per minute finds these lulls and swells in conversation. In the case of Obama's inauguration, on each minute, 25 percent of the tweets had an @ symbol with the exception of 3 consecutive minutes when the swearing-in took place. What we did not expect was that the number of tweets about the inauguration grew while the number of @ symbols dropped. See **Figure 1**.

Unlike Newton's Method across aggregated volume this works at Twitter's current scale and is more robust against different sampling techniques. Moreover, the more @ symbols we see on any given minute, the less important that moment should be.

A TABLE OF CONTENTS

Finally, when we turn to examining the text itself, a table of contents emerges. For this pick your favorite off-the-shelf information retrieval tool; mine is TF/IDF. What we want is a very specific table of contents, so we split the 90-minutes into 5-minute chunks, then find a highly salient slice term that is not salient in the other chunks. Looking back at the inauguration, this produces topic segments like: booing, Aretha, Yo-Yo Ma, anthem, and so on. Couple these words with our importance proxy, and we find the illuminating moments of Aretha Franklin's performance, Obama's speech, and the nation anthem to have greater importance than the other moments.

The signal of human activity here is clean and simple. Our analysis was done on a MacBook Pro using R and MatLab, without the need for large-scale assistance from a Hadoop cloud or Mechanical Turk.

Tweets can be used in social-multimedia research, when facilitated through human-centered research, to identify the shape of the related event. More than just text alone, the form of the conversational shadow has to account for the structure of tweets themselves. Hashtags, @ mentions, and other communication mechanism, both structured and ad-hoc, follow but likely do not mirror the visual content of the event itself. This link between two disparate data streams (online conversations and live events) provides rich opportunities for further investigation.

As we explore new approaches to better navigate, communicate, visualize, and consume events, our tools and research should firmly be based in the conversation structure, be it Twitter, Facebook, or Flickr, and be motivated by our interactions in everyday life, less we miss the game in the park.

Biography

David Ayman Shamma is a research scientist in the Internet Experiences group at Yahoo! Research. He researches synchronous environments and connected experiences both online and in the world. He designs and prototypes systems for multimedia-mediated communication, and develops targeted methods and metrics for understanding how people communicate online in small environments and at web scale. Ayman is the creator and lead investigator on the Yahoo! Zync project.

References

1. Cesar, P., Geerts, D., and Chorianopoulos, K. Social Interactive Television: Immersive Shared Experiences and Perspectives. *Information Science Reference*, 2009.

2. Shamma, D. A., Kennedy, L., and Churchill, E. F. Tweet the debates: Understanding community annotation of uncollected sources. In *WSM '09: Proceedings of the international workshop on Workshop on Social Media* (Beijing, China, 2009), ACM.

Ethics and Tactics of Professional Crowdwork

Paid crowd workers are not just an API call—but all too often, they are treated like one.

By M. Six Silberman, Lilly Irani, and Joel Ross

DOI: 10.1145/1869086.1869100

Faster, cheaper, smarter, and more efficient. These words might bring to mind the latest Intel ad, Moore's law, or hopes for cell phone processors—silicon, copper, and computation. These circuits, however, are not only embodied in semiconductors. Increasingly, masses of people sit at their keyboards computing the answers to questions artificial intelligence cannot. Programmers access these computing crowds through APIs or a GUI on a service such as Amazon Mechanical Turk (AMT). Working for a couple of dollars an hour, these anonymous computing workers may never meet the programmers who use them as part of their research and engineering efforts. Who are these mysterious workers? What kind of relationship do they have with the engineers who use them as human computation? How should we, as computing researchers, conceptualize the role of these people who we ask to power our computing?

In this article we discuss findings which suggest that these questions are increasingly important for those of us building the collection of technologies, practices, and concepts called human computation. We hope however that it will be understood as not only about human computation. Rather we hope to link the thus-far mainly technical conversations in human computation to discussions of engineering ethics that have gone on for at least forty years (see, e.g., Florman's Existential Pleasures of Engineering and Papanek's Design for the Real World). Here we offer insight into the practical problems crowdworkers face to ground these discussions in the current conditions of human computation.

Our research has focused on Mechanical Turk, a web platform that allows people ("requesters") to post information tasks called Human Intelligence Tasks ("HITs") for completion by other people ("workers" or "Turkers"), usually for a fee between one cent and a few dollars. Many businesses with large amounts of data use Mechanical Turk to create metadata and remove duplicate entries from their databases. Audio transcription and moderation of user-generated content on "Web 2.0" sites are other popular applications (see **Figures 1 and 2**).

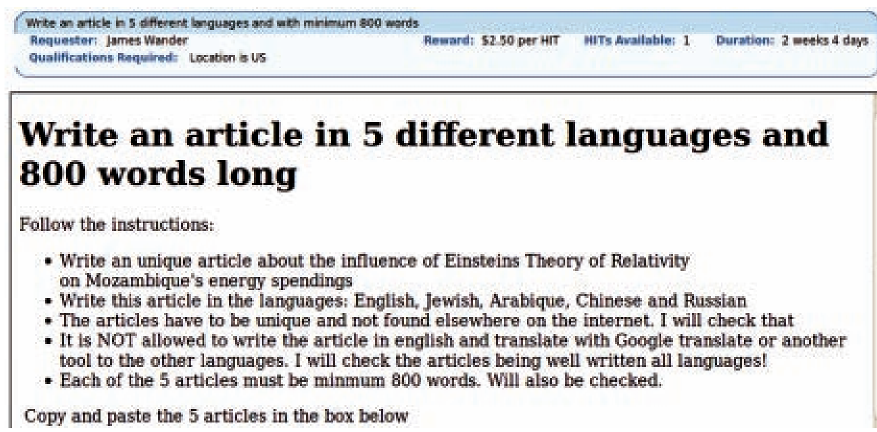
After a worker submits a HIT, the requester can decide to "accept" the work and pay the worker, or "reject" it and keep the work for free. The site keeps track of how often workers' submissions are accepted and rejected, and requesters use these rates to screen workers. When requesters reject work, they hurt workers' ability to get more work (especially more highly paid work) in the future. The frequencies with which requesters accept and reject work, however, are not made available to workers. This information asymmetry underlies many of the difficulties we discuss in this article.

With few exceptions, human computation research has focused on problems facing the requesters of human computation, and most investigations of workers have aimed to motivate better, cheaper, and faster worker performance. This makes sense sociologically: most researchers are requesters. Put simply, the requester's problem is to get good data from workers, quickly and without paying much. Workers, however, also have interesting and difficult practical problems.

Our last year studying Mechanical Turk from a worker point of view [1, 2] offers insights into opportunities for human computation researchers to think more broadly about the people who are crucial to the systems they build. We summarize the results of demographic studies of workers in Mechanical Turk and describe some of the problems faced by Turkers, as some workers call themselves. We present several projects, including one we built, that approach some of these problems. Finally, we explore open questions of interest to workers, requesters, and researchers.

Figure 1: The Mechanical Turk interface for HIT selection lets workers choose tasks to complete, ranging from audio transcription to content moderation.

Figure 2: Once a worker has selected a HIT, she is given instructions and a task to complete within a time limit.



THE CROWD AND ITS PROBLEMS

"I don't care about the penny I didn't earn for knowing the difference between an apple and a giraffe. I'm angry that AMT will take requesters' money but not manage, oversee, or mediate the problems and injustices on their site."

—*An anonymous worker*

Abstraction hides detail. The very abstraction that lets human computation researchers access thousands of workers in a click also renders invisible the practical problems faced by people in the crowdworking workforce. A number of surveys and active web forums offer glimpses behind the curtain where "artificial artificial intelligence" is made.

WORKING HARD FOR THE MONEY

The Mechanical Turk labor pool hosts a growing international population earning less than $10,000 per year, some of whom rely on Turking income to make basic ends meet. Ross et al. [2], extending work by Ipeirotis [3], present longitudinal demographic data on Mechanical Turk workers.

While Indian residents made up only 5 percent of respondents to a November 2008 survey, they comprised 36 percent of respondents to a November 2009 survey and 46 percent in February 2010, at which point American Turkers, formerly the majority, comprised only 39 percent of survey respondents. Many of these new Indian Turkers are young men earning less than $10,000 a year. Almost a third of Indian Turkers surveyed reported that they always or sometimes relied on their Turking income to "make basic ends meet." Between May 2009 and February 2010, the fraction of U.S. Turkers surveyed reporting reliance held steady at 13±1 percent.

Many Turkers see themselves as laborers doing work to earn money. In survey data collected in February 2009 (*n*=878), the most commonly reported motivation for doing HITs was payment: 91 percent of respondents mentioned a desire to make money. Turking to pass the time, in contrast, was mentioned by only 42 percent of respondents. February 2010 data (*n*=1,000) from Ipeirotis confirms the importance of money compared to other motivations, with most respondents reporting they do not do HITs for fun or to kill time. 25 percent of Indian respondents and 13 percent of U.S. respondents reported that Mechanical Turk is their primary source of income.

OCCUPATIONAL HAZARDS

What challenges face these professional crowdworkers? Several researchers have engaged workers by posting open-ended questions to Mechanical Turk—a sort of online interview to access a generally invisible population and see the world from their perspective. We have also conducted interviews of workers through Skype and participated in the forums where they share tips, talk about work, and virtually meet their coworkers. Turkers often advise one another on the occupational hazards of human computing:

Employers who don't pay: When workers submit work to employers through Mechanical Turk, they have no guarantee of receiving payment for their work. The site terms state that employers "pay only when [they're] satisfied with the results."

While this makes Mechanical Turk highly attractive to employers it leaves workers vulnerable to the whims of employers—or, just as likely, employers' evaluation software—judging the merit of their work. The amount of work often makes it impractical for employers to evaluate manually. Because employers hire hundreds or more workers at a time, they puzzle rejected workers with generic messages giving reasons for rejection, if they explain their decision at all. At worst, ill-intentioned employers post large batches of tasks with high pay, receive the work, and reject it as a way of obtaining free work. Such rejected work leaves workers feeling vulnerable, reduces their effective wage, and lowers their work acceptance rate.

Staying safe online: Mechanical Turk workers have to learn to identify illegitimate tasks to stay safe online. Administrator spamgirl on Turker Nation, a forum for workers, outlines tasks to avoid:

> *Do not do any HITs that involve: filling in CAPTCHAs; secret shopping; test our web page; test zip code; free trial; click my link; surveys or quizzes (unless the requester is listed with a smiley in the Hall of Fame/Shame); anything that involves sending a text message; or basically anything that asks for any personal information at all—even your zip code. If you feel in your gut it's not on the level, IT'S NOT. Why? Because they are scams…*

The discussion that ensued identified malware, sale of personal information and wage theft as risks workers face choosing among jobs.

"Why is there no control?" Hit by several of the problems described above, a4x401 offered a newcomer's frustrated perspective with the worker side of human computation:

> *Being a newbie and having relatively decent PC skills, I have been checking all this stuff out and am somewhat upset [about] the things that I have discovered! It's no wonder that people don't trust the requesters, yes I did some of those HITs that one should not do and found myself having to repair my PC and remove some pop-ups. After having done that I really got into checking out the program and realized that it's too easy to manipulate it due the fact that work can be rejected after it's finished but the work is still done. All [a requester] has to say is "not to our satisfaction"!!!!! The other way is to just leave the HITs open; you still collect your work but don't have to pay! My favorite part is HITs that are way too complicated to complete in the time frame allowed! Why is there no control on any of this stuff?* [Edited for punctuation and spelling.]

He echoes experiences many report on worker forums and in research surveys. Workers report trying to contact Amazon staff but receiving little response.

Costs of requester and administrator errors are often borne by workers: When a requester posts a buggy task or a task with inadequate instructions, they often don't get the responses they want from workers and reject the work. One worker wrote:

> *I would like to see the ability to return a HIT as defective so it dings the requester's reputation and not mine. Let's face it, if I'm supposed to find an item for sale on Amazon but they show me a child's crayon drawing…there really needs to be a way to handle that without it altering my numbers.*

Similarly, occasionally requesters will post a task with a prohibitively short time limit, and the task expires before workers can complete it. This lowers workers' effective wage and affects the worker's reputation statistics rather than the requester's.

At present, largely owing to requester and administrator unresponsiveness, workers can do little to improve the conditions of their tasks. Unsurprisingly, some have expressed interest in a more relationship-oriented approach to distributing work. One Turker wrote:

> *We the Turks, in a world that requires productivity in working together, will work honestly and diligently to perform the best work we can. The Requestors, in turn, will provide useful work and will pay us fairly and quickly, providing bonuses for especially good work. The goal is to create a working environment that benefits us all and will allow us the dignity and motivation to continue working together.*

These are some of the more prevalent phenomena we've encountered. For more, see turkernation.com and mturkforum.com, especially their "Suggestions" boards.

APPROACHES TO WORKER PROBLEMS

Software tools exist, some built by Turkers, that attempt to help Turkers manage these problems. Many are client-side scripts that add functionality to the Mechanical Turk interface. At least one platform aims to compete with Mechanical Turk.

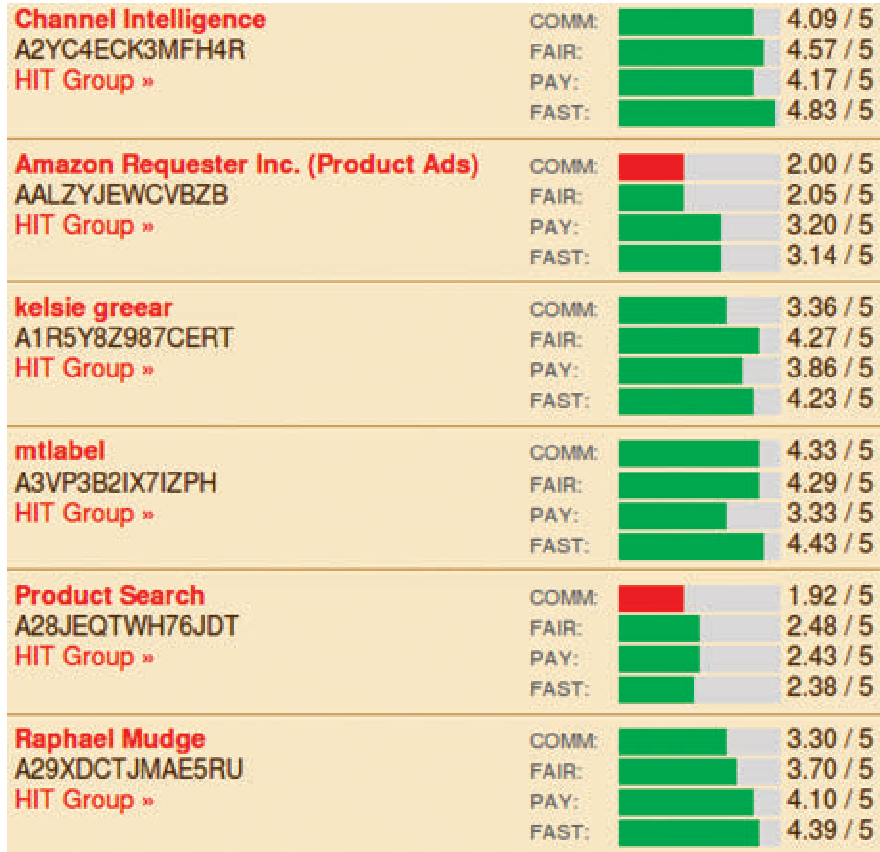
Augmenting Mechanical Turk from the outside: Workers and requesters have made a number of Turking tools, including a list of all requesters, a script for recording your own worker history (not preserved by Mechanical Turk, but useful for tax purposes), and a client-side script to hide HITs posted by particular requesters.

Motivated by the problems above, we built Turkopticon (turkopticon.differenceengines.com) in 2008, a database-backed Firefox add-on that augments Mechanical Turk's HIT listing. The extension adds worker-written reviews of requesters to the interface (see **Figures 3, 4, and 5**); the next version will compute effective wage data for HITs and requesters. Some Turkers have been enthusiastic about Turkopticon. One early adopter posted to Turker Nation, "if you do not have this, please get it!!!! it does work and is worth it !!"

This was a proud moment for us,

Figure 3: Turkopticon augments the MTurk interface with Turker feedback using a Firefox extension.

Figure 4: Turkopticon is a feedback platform for Turkers. Turkers can rate requesters on communicativity, generosity, fairness and promptness.



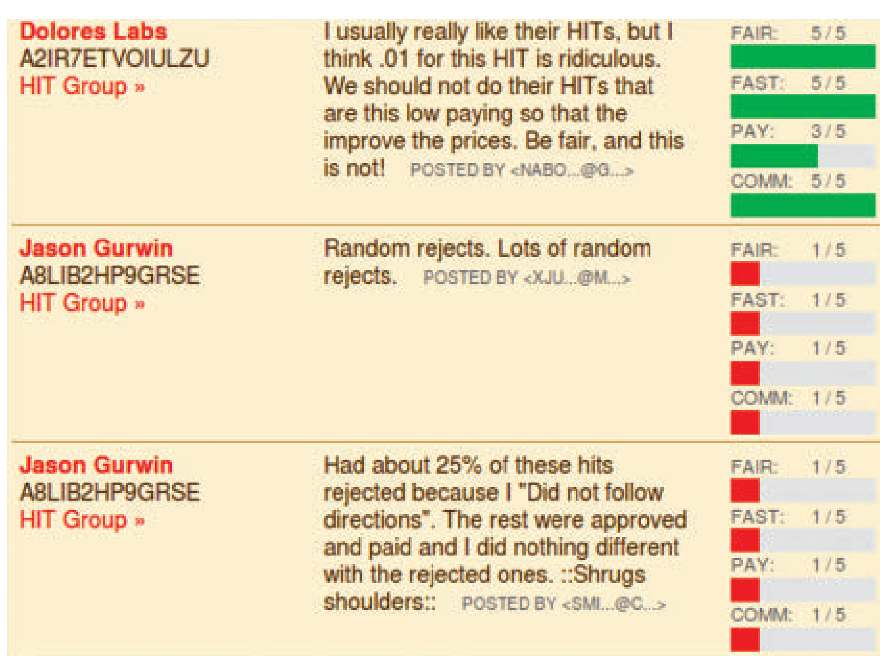
and we have attempted to respond to feature requests and provide support for new users. Turkopticon users have contributed over 7000 reviews of over 3000 requesters, but the user base has remained very small, especially compared to the total number of Turkers. Relatedly, some Turkers have pointed out that a third-party review database is no subsitute for a robust, built-in requester reputation system.

Building alternative human computation platforms: CloudCrowd, launched in September 2009, aims to provide a "worker-friendly" alternative to Mechanical Turk. In a post to mTurk Forum, CEO Alex Edelstein writes that CloudCrowd will offer "a more efficient [worker] interface," payment through PayPal (allowing workers to collect currencies other than USD and INR, the only choices for Turkers), and "credibility" ratings (in place of acceptance rates as in Mechanical Turk) as the measure of worker quality.

Kochhar et al., in a paper at HCOMP 2010 [4], documented the success of a relationship-oriented approach to distributing work in the design of a "closed" large-scale human computation platform.

Offering workers legal protections: Alek Felstiner has raised the question of legal protections for crowdworkers, asking, "what [legal] responsibilities, if any, attach to the companies that develop, market, and run online crowdsourcing venues?" In his working paper [5] he explores the difficulties that arise in the application of traditional employment and labor law to human computation markets.

Figure 5: Turkers can provide free text feedback to explain their ratings.



OPEN QUESTIONS

The projects listed above are tentative steps toward addressing the problems facing Turkers and developing a richer understanding of the structure and dynamics of human computation markets. Many questions remain, including: How does database, interface, and interaction design influence individual outcomes and market equilibria?

For example, how would the worker experience on Mechanical Turk be different if workers knew requesters' rejection rates, or the effective wages of HITs? This has been explored in online auctions, especially eBay, but only tentatively in human computation (e.g., [6], which examines task search).

Another question is: What are the economics of fraudulent tasks (scamming and spamming)?

That is, how do scammers and spammers make money on Mechanical Turk, and how much money do they make? Work in this thread might draw on existing research on the economics of internet fraud (e.g., [7]) and could yield insights to help make human computation markets less hospitable to fraudsters.

A third question is: What decision logics are used by buyers and sellers in human computation markets?

We might expect workers to minimize time spent securing payment on each task, even if this means providing work they know is of low quality. Some workers do behave this way. We have found, however, that workers seem more concerned with what is "fair" and "reasonable" than with maximizing personal earnings at requester expense. The selfish optimizers that populate the models of economic decision-making may not well describe these "honest" workers, although as noted in [8] they can perhaps be extended to do so. So how do differently motivated actors in human computation markets shape market outcomes, and how can this knowledge shape design?

Finally, we can ask: What's fair in paid crowdsourcing?

Economists Akerlof and Shiller, in their 2009 book *Animal Spirits: How Human Psychology Drives the Economy, and Why It Matters for Global Capitalism*, argue that "considerations of fairness are a major motivator in many economic decisions" that has been overlooked in neoclassical explanations that assume people act rationally: "while...there is a considerable literature on what is fair or unfair, there is also a tradition that such considerations should take second place in the explanation of economic events" (pp. 20, 25).

At public events we have heard Mechanical Turk requesters and administrators say tasks should be priced "fairly," but fairness is difficult to define and thus to operationalize. The concept of a reservation wage—the lowest wage a worker will take for a given task—as discussed in [9] is useful but not definitive: the global reach of human computation platforms complicates the social and cultural interpretation of the reservation wage.

The question of fairness links interface design to market outcomes. If considerations of fairness are key to explaining economic decision making, but fairness is constructed and interpreted through social interaction, then to understand economic outcomes in human computation systems we need an understanding of these systems as social environments. Can systems with sparse social cues motivate fair interactions? Human computation and Computer Supported Cooperative Work may have much to learn from one another on these topics.

LOOKING FORWARD

This review of workers' problems should not be mistaken as an argument that workers would be better off without Mechanical Turk. An exchange in late 2009 on the Turker Nation forum makes the point concisely:

xeroblade: *I am worried that Amazon might just shut the service down because it's becoming full of spammers.*
jml: *Please don't say that :(*

With Mechanical Turk, Amazon has created work in a time of economic uncertainty for many. Our aim here is not to criticize the endeavor as a whole but to foreground complexities and articulate desiderata that have thus far been overlooked. Basic economic analysis tells us that if two parties transact they do so because it makes them both better off. But it tells us nothing about the conditions of the transaction. How did the parties come to a situation in which such a transaction was an improvement? When transactions are conditioned by the intentional design of systems, we have the opportunity to examine those conditions.

Human computation has brought Taylorism—the "scientific management" of labor—to information work. If it continues to develop and grow, many of us "information workers" may become human computation workers. This is a selfish reason to examine design practices and workers' experiences in these systems. But the underlying question is simple: are we, as designers and administrators, creating contexts in which people will treat each other as human beings in a social relation? Or are we creating contexts in which they will be seduced by the economically convenient fiction alluded to by Mechanical Turk's tagline, "artificial artificial intelligence"—that is, that these people are machines and should be treated as such?

Biographies

M. Six Silberman is a field interpreter at the Bureau of Economic Interpretation. He studies the relation between environmental sustainability and human-computer interaction. His website is wtf.tw.

Lilly Irani is a PhD candidate in the Informatics department at University of California-Irvine. She works at the intersection of anthropology, science and technology studies, and computer supported cooperative work.

Joel Ross is a PhD candidate in the Informatics department at University of California-Irvine. He is currently designing games to encourage environmentally sustainable behavior.

References

1. Silberman, M. S., *et al.* Sellers' problems in human computation markets. In *Proceedings of HCOMP 2010*.
2. Ross, J., et al. Who are the crowdworkers? shifting demographics in Mechanical Turk. In *Proceedings of CHI 2010*: 2863-2872.
3. Ipeirotis, P. Mechanical Turk: the demographics. http://behind-the-enemy-lines.blogspot.com/2008/03/mechanical-turk-demographics.html.
4. Kochhar, S., et al. The anatomy of a large-scale human computation engine. In *Proceedings of HCOMP 2010*.
5. Felstiner, A. Working the crowd: employment and labor law in the crowdsourcing industry. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1593853.
6. Chilton, L., et al. Task search in a human computation market. In *Proceedings of HCOMP 2010*.
7. Franklin, J., et al. An inquiry into the nature and causes of the wealth of internet miscreants. In *Proceedings of CCS '07*: 375-388.
8. Jain, S. and D. Parkes. The role of game theory in human computation systems. In *Proceedings of HCOMP 2009*.
9. Horton, J. J. and L. Chilton. The labor economics of paid crowdsourcing. arXiv:1001.0627v1 [cs:HC], 2010.

Games for Extracting Randomness

Two computer scientists have created a video game about mice and elephants that can make computer encryption properly secure—as long as you play it randomly.

By Ran Halprin and Moni Naor

DOI: 10.1145/1869086.1869101

People have been sending messages in code almost since writing began—and others have been trying to crack those codes. The study of the design and breaking of codes is known as cryptography. With the advent of computers, the scope of cryptography has expanded significantly to now deal with methods for ensuring the secrecy, integrity, and functionality of computer and communication systems in face of an adversary.

Randomness is essential for addressing the main problems of cryptography (encryption, identification and authentication). When designing a code, one has to select its architecture—the mathematical method behind it. It is commonly assumed in cryptography that your adversary will know or find out the architecture of your code. The assumption is referred to as Kerckhoffs' principle. The part assumed to be unknown by the adversary, at least initially, is called the key. It should be selected from a large space of possible keys.

THE PROBLEM WITH RANDOMNESS

When designing random algorithms and cryptographic systems, the availability of a source of pure randomness is assumed, but such a perfect source of randomness is not known to exist. The most common solution is to find a source assumed to be somewhat random (often denoted "entropy source"), or at least unpredictable from the point of view of an adversary, and then apply some hash (mixing) function to it. The randomness source usually consists of a combination of information assumed to be unknown to an adversary (e.g., components manufacturer ID) combined with information assumed to be difficult to predict (e.g., exact system time or mouse position or movement). This process is called randomness extraction.

Entropy is a measure of the uncertainty of a source or random variable. The most well known variant, the Shannon entropy, measures the expected length of the best encoding. For the purposes of randomness extraction, especially for cryptography, a more important measure is the min-entropy, the measure the probability of guessing the source, that is, the most common value. For a random variable *X* with *n* possible values the Shannon entropy is: $H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$. The min-entropy is $H_\infty(X) = \min_{x\in\{0,1\}^n} \log_2 \frac{1}{\Pr[X=x]}$. It is known that $0 \le H_\infty(X) \le H(X) \le \log_2 n$

While in principle random extraction is a good approach, its implementation is often problematic and not based on a rigorous approach. There have been some well known failures of this type. Early in the history of the web, Goldberg and Wagner attacked Netscape's implementation of the SSL protocol—the cryptography designed to protect such things as secure internet credit card transactions. The attack was based on the Randomness Generator's entropy sources and refresh rate **[10]**.

Even more than a decade later, security vulnerabilities are still found in pseudo-random generators (PRG). For example, a bug in the pseudo-random generator of the Debian Linux operating system (including the popular Ubuntu Linux distribution) caused the generated cryptographic keys to be insecure, since they accidentally relied on very little entropy. This bug was only discovered in May 2008, two years after being introduced into the system **[20]**.

Entropy collection is the first critical step in generating pseudo-randomness. Today, randomness generators use one or more of three types of entropy collection. One is background gathering—constantly collecting data from the operating system such as changes in memory or mouse position. This method is used for example by Windows and Linux. One problem here is that usage patterns might be predictable. The mouse might be frequently in the lower-left corner of the screen, for example, which makes it non-random.

An alternative is hardware gathering. The system observes external chaotic systems such as cloud patterns, electrical circuits, atmospheric noise or even lava lamps [**3, 11, 14**]. Such systems have large overheads, are physically complex and are therefore expensive to create and maintain. The third system is direct gathering. The user is requested to hit the keyboard or wiggle the mouse randomly for a period of time. PGP is a well known example of direct entropy request. The main problem with this method is that humans click the keyboard or move the mouse in relatively predictable patterns. Even worse, the casual user will be bored with requests for entropy and might perform very similar patterns every time they are prompted.

The issue of obtaining good randomness is not one of "send and forget." We do not want the system to have the property that a single leak compromises the system forever. For instance, HTTP session IDs are assumed to have randomness and should be hard to guess. Therefore we need methods to refresh our randomness periodically.

HUMANS AND RANDOMNESS

It is generally accepted that sequences and numbers generated by humans are far from being truly random. Ask someone to choose a number between 1 and 10 and they will usually choose 7. For the range 1 to 20, they usually choose 17. Other human biases in randomness assessment are well researched. There is the "hot hand," the tendency to believe that a winning streak will usually continue. Its inverse is the "gambler's fallacy," the belief that after a losing streak, the next attempt is more likely to be a success, and the related "flip bias," which is the all-too-human tendency to believe that a 0 is likely to be followed by 1, or, in roulette, that red will be followed by black, and vice versa. These beliefs have been shown, statistically, to be fallacies) [**4, 6, 18**].

> **"In a random binary string of length 100 either the string 111111 or the string 000000 should appear with probability 95 percent, but a human finds that hard to believe."**

"Flip bias" has been shown to exist in randomness generation as well [**16**]. Figurska *et al.* showed that humans asked to generate a string of digits tend to choose successive identical digits with probability 7.58 percent instead of 10 percent [**7**]. Thus they would produce sequences such as 5, 7, 9, 9, 4, 2 less frequently than they should. Due to flip bias, humans expect (and generate) sequences that contain shorter repeating sequences than true randomness would. For example, in a random binary string of length 100 either the string 111111 or the string 000000 should appear with probability 95 percent, but a human finds that hard to believe and would assess strings that contain them as non-random.

An interesting twist in the study of humans and randomness emerged when Budescu and Rapoport demonstrated that under some conditions humans can attain near-randomness [**15**]. It happens when they are playing the zero sum competitive game "matching pennies." In this game, each player chooses a red or a black card. The first player wins if each player chooses a different card and the second player wins if the cards are identical. When playing this game over many rounds, the players have an incentive to select cards randomly, so that their opponent cannot predict their choice—and human choices in matching pennies are much closer to random than when they actually try to simulate a random process.

Newer experiments with monkeys and bananas, and humans and other rewards, seem to support the claim that higher primates are able to behave almost randomly while participating in competitive games [**2, 8, 9**].

IT IS ONLY HUMAN TO BE BIASED

While better than expected, human's gameplay patterns are still not perfectly random. Eliaz and Rubinstein produced a slight variant on matching pennies, in which the first player is called the "misleader" or "hider," whose specific task is to choose a card that would fool his opponent. The opponent is called the "guesser" or "seeker," who has to deduce from past behavior the card that the opponent has chosen.

Mathematically, the game is identical to matching pennies. The only difference is psychological. In matching pennies, the players have exactly identical roles. In the variant, one has become a hider, the other a seeker. Yet that change in psychology seems to affect the outcome. A guesser seems to have a slight empirical advantage over a misleader [**5**]. Specifically, guessers have a winning rate of up to 0.542 (rather than the expected 0.5).

The authors attribute this advantage mostly to the natural thought pattern used by guessers (trying to think like their opponent rather than think the opposite from their opponent), and their strategic framing (they are ordered to think where the other player hides, while the misleaders are not told to think where they will be looked for). They were able to eliminate this bias by using a variation of the game in which both players turned into guessers. (Player 1 chooses 'a' or 'i', player 2 chooses 's' or 't'. Player 1 wins if the word created is "as" or "it," player 2 wins if the word is "is" or "at"). This time, symmetry returned and the long-term results were 50 percent for both players.

A more disturbing bias in the context of our work is the tendency of winning guessers to repeat their choice (60.7 percent) and the tendency of winning misleaders to change their choice after success (57.7 percent). It is not clear how much this effect reproduces in more complex (non-binary) games.

A NEW TYPE OF ENTROPY SOURCE

Knowing that humans posses the capacity to generate randomness (at least to some extent) when they are playing some games, we can predict how random they will be. In order to extract entropy—that is, randomness—from human gameplay, we would like a game that encourages humans to play as randomly as possible.

Game theory often discusses games with mixed strategy, but most recreational games actually played by people have a deterministic optimal strategy, i.e., a finite sequence of moves that prevents losing in the game, or in the case of randomness-based games (such as poker) minimize the risk of losing. This includes also single-player video games such as *Pac-Man* or *Super Mario Bros.* These games rely on the player's skill in finding and performing winning sequences. Such games are obviously less useful for our entropy extraction, as play is usually far from random (although some entropy can be found in probably any gameplay sequence), so we want to identify games that encourage humans to be as random as possible.

The game matching pennies has two drawbacks. The first is that while it encourages humans to be random, it only generates one raw bit of data (which means at most one bit of entropy) per round, and a modern random key is at least 128 bits long.

The other problem is that we need a game that is at least somewhat interesting so that it entertains players long enough for them to generate sufficient randomness. Matching pennies is a rather boring game (Rapoport and Budescu had to pay their student test subjects to play it).

Another issue we need to tackle is what to do with the fact that people's choices are still not perfectly random and may in fact be quite far from random. These biases can be fixed by using randomness extractors (to be defined later) and the only assumption we need is that no single pattern is too dominant. That is, the probability that the adversary can guess perfectly all the opponent's moves is bounded by some small value—an assumption on the min-entropy of the player's actions.

"Some players use the natural tactic of repeating the same moves over and over from round to round [which prevents true randomness]."

OF MICE AND ELEPHANTS

We chose to study a natural extension of matching pennies called hide-and-seek. In the classic version of hide-and-seek, one player (the hider) chooses a cell on a vector or grid. The second player (the seeker) also chooses a spot on the same grid. The hider wins a point if the seeker chose a different cell, and the seeker wins a point if they both chose the same cell.

Thus unadorned, the hide-and-seek game does not seem to be much more entertaining than matching pennies. To make the game more enjoyable so that people will want to play it for the required length of time, we used lively mouse and elephant characters, we renamed it "mice and elephant" and incorporated animations and sound effects. An alternative might have been to develop the game of Battleship, with "fleets" secretly positioned on squared paper, and each fleet being shot at in turn by the opponent. As far as we know, no one has ever had to pay nine-year-olds to play Battleship.

Mice and elephants, however, is played on-screen, on a 512×256 pixel board which fits the screen nicely and produces 9+8=17 raw bits of entropy per user click. (512=29, 256=28). See **Figure 1**. The human player controls mice (1 in the first round, 2 in the second and third, 3 in the fourth and fifth, etc.), and the computer controls the elephant (the game is available online at http://math166-pc.weizmann.ac.il). In each round, the hider positions the mice on the grass, and the seeker positions the elephant. After both players perform their moves, a short animation sequence shows the elephant falling on the spot where it was placed. The game ends when the seeker's elephant "crashes" one of the hider's mice and flattens it. The objective of the hider is to survive as many rounds as possible, while the objective of the seeker is to catch a mouse as early as possible, ending the game.

The progression in the number of mice was chosen mainly to make the game more interesting. The increasing probability for losing in each round helps build tension in the game, especially when the player approaches his previous highest round. An important effect of this choice is that it allows good players to introduce more entropy into the system. A more random player will generally reach higher rounds, as the opponent will find it harder to predict his behavior.

Players cannot position the mice on top of each other, and the game field also has obstacles, which are positioned in popular click locations to prevent repetitive behavior.

One problem we've seen is some players use the natural tactic of repeating the same moves over and over from round to round. A simple solution was having the elephant, with high probability, choose its moves uniformly from the player's recent history of moves (once the player builds such a history—before that the moves are always random). This ensures that playing a certain region too often will cause the player to be caught faster.

The game was released to the public as an experiment. The purpose was to study how people played it, and specifically how predictable (and therefore not random) they are. The experiment ran online for several weeks during which it was advertised on various internet forums, to friends and family, and in several academic gatherings. It is important to note that most subjects were not aware of our objective and were only instructed to attempt to mislead the elephant and survive as many rounds as possible. For analysis of the results of this experiment, see the full publication **[12]**.

SECURE PSEUDO-RANDOM GENERATION

To fully specify the system, we need to describe how we obtain randomness

Figure 1: Positioning mice in the game mice and elephants generates randomness. (Graphic courtesy of Anat Iloni).

from the gameplay. We construct a fully functional PRG system which can be tested online http://math166-pc.weizmann.ac.il or downloaded from http://www.neko.co.il/MAE-offline.html. The system lets the user play and collects the gameplay data. This data is not perfectly random, but should have sufficient entropy. Using a randomness extractor we create from this data an almost-random sequence, which we then use to seed a cryptographic pseudo-random generator.

RANDOMNESS EXTRACTORS

What does it mean to be far from uniform or close to uniform? For this we use the measure known as statistical distance: Two distributions P and Q over the same domain T are **statistically ε-close** if: $\frac{1}{2}\sum_{x\in T}|P(x)-Q(x)|\leq \varepsilon$.

The goal of randomness extractors is to start with a source that is far from uniform and transform it into one that is close to uniform. The earliest discussion of using "weak" random sources in order to get full randomness was made by von Neumann, who considered the case of a biased source that outputs a 0 with probability p and 1 with probability (1–p) **[19]**. Uniform randomness can be extracted from this source by considering pairs of bits: If the pair is 01 we output 0. If the pair is 10 we output 1. If the two bits are identical, we ignore this pair. The resulting sequence is perfectly uniform, regardless of the value of p.

In recent times the theory of randomness extractors has been thoroughly investigated. Efficient methods have been found to extract randomness from sources without a known structure, where the only requirement is high min-entropy and some random "seed."

A (k, ε)-extractor is a function Ext: $\{0,1\}^n$ x $\{0,1\}^d$ — $\{0,1\}^m$ such that for every distribution X on $\{0,1\}^n$ with $H\infty(X) \geq k$, the distribution $Ext(X, s)$ (where $s \in_R \{0,1\}^d$, i.e., the seed s is d bits randomly chosen from the uniform distribution over $\{0,1\}$) is statistically ε-close to the uniform distribution on $\{0,1\}^m$.

While extractors can be used in cryptographic systems, the seed of extractor might become known to the adversary, such as when the system's randomness is temporarily compromised. If the extractor's output is still close to random even when the seed is provided, we call the extractor "strong."

A (k, ε)-strong extractor is a function Ext as above, where the distribution $Ext(X, s)$ followed by s is statistically ε-close to the uniform distribution on $\{0,1\}^{m+d}$. Using a strong extractor implies that the seed is reusable. In many applications, it is therefore possible for this seed to be hard-wired into the system once in its creation. There are good constructions of strong extractors, and a fairly simple but quite powerful one can be based on random linear functions over finite fields. For more about extractors see Shaltiel's survey **[17]**.

CRYPTOGRAPHIC PSEUDO-RANDOM GENERATORS

A cryptographic pseudo-random generator is an efficiently computable function that, given a random seed, creates a longer random looking sequence of bits. This sequence should look indistinguishable from random for any observer who computation power is limited. In general, for every feasible test we want the probability that the output of the generator will "pass" the test to be close to the probability that a truly random string passes the test. The precise definition of Cryptographic PRGs is beyond the scope of this paper (for more details see, for example, the textbook by Katz and Lindell **[13]**.

PRGs are extremely important in Cryptography, as they imply that a relatively short random seed can supply all the randomness needed for a system without significant security loss relative to a one-time pad.

While a PRG can be used to expand a short random sequence into a longer sequence, there are still considerable security vulnerabilities for real PRG systems. If an adversary manages to retrieve the seed used to generate the sequence, the entire sequence becomes predictable. Even partial information about the seed could be used by an adversary to perform educated guesses, and even short pieces of the generated sequence could be used to verify these guesses.

In order to address this problem, Barak and Halevi introduced a robust pseudo-random generator **[1]**. A robust PRG has two input functions: *refresh()* that refreshes its entropy, and an output function *next()* that returns a piece of the pseudo-random sequence. A robust PRG should satisfy the following properties:

Forward security: assume the adversary learns the internal state of the system at a certain point in time. The past outputs of the system generated prior to the break-in, should still be indistinguishable from random to the adversary (naturally, this means the past states should not be reconstructable).

Backward security (break-in recovery): assume the adversary learns the current internal state. Following the next "refresh" (after the break-in), all outputs should be indistinguishable from random to that adversary.

Immunity to adversarial entropy: assume the adversary gains complete control over the refresh entropy (but has no knowledge of the internal state of the system which has been previously refreshed). The output of the system should still be indistinguishable from a random sequence to that adversary.

A major advantage of robust PRGs is the ability to combine independent

"Under some conditions humans can attain near-randomness. It happens when they are playing the zero sum competitive game 'matching pennies.'"

sources of entropy for a proven increase in security. This possibility can also be seen as a motivation for the current work and for finding other sources of entropy as different as possible from those in use today.

Barak and Halevi show how to construct a provably robust PRG from any cryptographically secure PRG.

RANDOMNESS THROUGH FUN

Using the results from the experiment, we estimated how much randomness people were putting in this game. Based on the estimation of the min-entropy, we found that 39 mouse clicks, are sufficient to seed a PRG with 128 bits which are 2128-close to random. That is, assuming the game was played with enough randomness, the resulting string has high min-entropy. The results of extraction are inserted as a seed into an implementation of the Barak-Halevi system that we created, which can be reseeded by gameplay whenever the user wishes.

How do we know whether the result is indeed random? The disappointing answer is that we cannot really know. It is very difficult to assert that the results are indeed random due to the universal requirement of cryptographic pseudo-randomness—that it fools all possible tests. Therefore, one possibility for a sanity check is to use a standard test of randomness. One such randomness assessment suite is known a DIEHARD. In order to test our system, we let it generate millions of keys (with the clicks played in the game as a single randomness source) and ran them through the DIEHARD suite, which gave it very good results, similar to Microsoft Windows' Cryptographic Randomness Generator.

Our main conclusion from this work is that using human gameplay as a randomness source is a valid option for generating secure randomness, whether by itself when other forms of randomness are not available, or by combining it with other entropy sources. We also learned of the power of games as a motivator to participate in experiments, as we collected more information than is customary in such research and more easily. And—always useful given research budgets—we did not have to pay our participants.

"Using human gameplay as a randomness source is a valid option for generating secure randomness"

The downside is that we don't really know who the group of participants was, their demographics, and how accustomed they are to playing computer games.

Biographies

Ran Halprin holds a MSc. in computer science from the Weizmann Institute of Science. His research interest is in human computation and its applications, mainly in cryptography.

Moni Naor is a professor at the Department of Computer Science and Applied Mathematics at the Weizmann Institute of Science and the incumbent of the Judith Kleeman Professorial Chair. His research interest is in foundations of computer science, especially cryptography.

References

1. Barak, B. and Halevi, S. 2005. A model and architecture for pseudo-random generation and applications to /dev/random. *ACM Conf. Computer and Comm. Security*, 203˝U–212.
2. Barraclough, D., Conroy, M., and Lee, D. 2004. Prefrontal cortex and decision making in a mixed-strategy game. *Nature Neuroscience 7*, 4, 404–410.
3. Bernstein, G. and Lieberman, M. 1990. Secure random number generation using chaotic circuits. Circuits and Systems, *IEEE Transactions on 37*, 9, 1157–1164.
4. Burns, B. and Corpus, B. 2004. Randomness and induction from streaks: "gambler's fallacy" versus "hot hand". *Psychonomic Bulletin and Review 11*, 179–184.
5. Eliaz, K. and Rubinstein, A. 2008. Edgar Allen Poe's Riddle: Do Guessers Outperform Misleaders in a Repeated Matching Pennies Game? http://arielrubinstein.tau.ac.il/papers/poe.pdf.
6. Falk, R. and Konold, C. 1997. Making sense of randomness: Implicit encoding as a basis for judgment. *Psychological Review 104*, 2 (April), 301–318.
7. Figurska, M., Stańczyk, M., and Kulesza, K. 2008. Humans cannot consciously generate random numbers sequences: Polemic study. *Medical Hypotheses 70*, 1, 182–185.
8. Flood, M., Lendenmann, K., and Rapoport, A. 1983. 2×2 Games played by rats: different delays of reinforcement as payoffs. *Behav Sci 28*, 65–78.
9. Glimcher, P., Dorris, M., and Bayer, H. 2005. Physiological utility theory and the neuroeconomics of choice. *Games and Economic Behavior 52*, 2, 213–256.
10. Goldberg, I. and Wagner, D. 1996. Randomness and the netscape browser. *Dr. Dobb's Journal*, 66–70.
11. Haahr, M. Random.org: true random number service. http://www.random.org.
12. Halprin, R. and Naor, M. 2009. Games for extracting randomness. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, article 12, California.
13. Katz, J. and Lindell, Y. 2008. *Introduction to modern cryptography.* Chapman & Hall/CRC.
14. Noll, L., Mende, R., Sisodiya, S., et al. 1998. Method for seeding a pseudo-random number generator with a cryptographic hash of a digitization of a chaotic system. US Patent 5,732,138.
15. Rapoport, A. and Budescu, D. 1992. Generation of random series in two-person strictl competitive games. *Journal of Experimental Psychology: General 121*, 3, 352–363.
16. Rapoport, A. and Budescu, D. 1997. Randomization in individual choice behavior. *Psychological Review 104*, 1, 603–617.
17. Shaltiel, R. 2004. Recent developments in explicit constructions of extractors. *Current Trends in Theoretical Computer Science: The Challenge of the New Century.*
18. Tversky, A. and Gilovich, T. 2004. The "Hot Hand": Statistical Reality or Cognitive Illusion? Preference, Belief, and Similarity: Selected Writings.
19. von Neumann, J. 1951. Various techniques used in connection with random digits. *Applied Math Series 12*, 36–38.
20. Weimer, F. New openssl packages fix predictable random number generator. http://lists.debian.org/debian-securityannounce/2008/msg00152.html.

PROFILE

Luis von Ahn
ReCaptcha, Games With a Purpose

BY ROBERT J. SIMMONS

DOI: 10.1145/1869086.1869102

In 2002, Luis von Ahn had a crazy idea: he wanted to use an online computer game to perform the tedious task of "image labeling," or identifying pictures and assigning them textual descriptions. The problem with crazy ideas, of course, is that they sound a little crazy, and so it's hard to get other people to believe in them. Von Ahn's first paper about the game was rejected from the ACM Conference on Human Factors in Computing Systems (also known as "CHI") with some of the lowest reviewing scores possible.

Uninterested in taking "no" for an answer, von Ahn spent several months realizing his idea as a web-based game called the *ESP Game*. In the *ESP Game*, a player logs into a web site and is paired up with another random player. The two players both see the same image and are asked to write words that describe it. As soon as the two players write the same word, they win the round, and a new image appears. Behind the scenes, the matched words are then used as labels for the image. In the first four months of its operation, the *ESP Game* generated more than a million labels for more than a quarter of a million images.

The game is available to anyone to play at Games with a Purpose: http://www.gwap.com.

His paper, now with a very significant proof-of-concept, cruised to acceptance at the 2004 CHI conference with rave reviews.

Experiences like this have made von Ahn accept the fact that it sometimes takes extra work to prove everyone else wrong. Not only do ideas like the *ESP Game* take months or years to develop, but the development of those ideas requires collaboration across many areas of computer science. When it comes to developing games, it's not enough for the human computation aspects to work correctly. The game has to be fun, easy-to-use, and capable of handling potentially millions of people trying to play simultaneously.

Photo by Joshua Franzos.

In the first four months of its operation, the ESP Game generated more than a million labels for more than a quarter of a million images.

While the *ESP Game* has certainly come into its own, von Ahn is perhaps better known for creating Captcha, a system that helps distinguish humans from bots online, and its spin-off ReCaptcha.

ReCaptcha works by showing people pictures of words that a computer had trouble recognizing as text and asking people to type it. Ticketmaster likes it because it helps ensure that the company is interacting with a person and not a scalper's computer program. Von Ahn likes it because, by typing in the word that appears in a difficult-to-read image, software that he and his team designed can reliably turn pictures of a book into words. However, not-so-good people in the real world, like scalpers and organized crime syndicates, want access to the resources that ReCaptcha protects, whether free email accounts, concert tickets, or Craigslist postings, and the results are sometimes disturbing. While von Ahn says he's never felt threatened directly, it did hit a little too close to home when people tried to turn off some of ReCaptcha's servers by breaking into a Los Angeles facility where they were stored.

Von Ahn spends much of his time at Carnegie Mellon University teaching undergraduates and advising graduate students. Teaching, in particular, is a frequent topic on his aptly-named web site "Luis von Blog" (http://vonahn.blogspot.com), another place where he is unafraid of controversy.

One blog post, where von Ahn weighed the advantages of outsourcing graduate students to third-world countries, came with the following disclaimer: "100 percent of my PhD students are working on projects of their own choosing." Even though they have a well-known advisor, von Ahn says that the graduate students he advises have not found it difficult to differentiate their work from his. Most of them identify with different areas of computer science and publish in different conferences.

It helps that von Ahn is uninterested in fitting into a field, whereas many other researchers find it important to concentrate on a particular area of computer science. The CMU professor thinks that's just crazy, even if holding that belief implies, as he says, that "everyone is crazy but me."

Running the Turk

Interview with Amazon.com vice president Sharon Chiarella and PR manager Kay Kinton

To find out how Amazon.com runs its marketplace for crowdsourced labor, we spoke to the vice president at the company responsible for it.

Interviewed by Nelson Zhang

DOI: 10.1145/1869086.1869103

Mechanical Turk, part of Amazon's Web Services, is an online marketplace into for crowdsourced labor. It allows users, called "providers," to complete human intelligence tasks, or HITs, offered by "requesters." The requesters pay the providers an agreed-upon price for each completed HIT. Sharon Chiarella is vice president of Amazon.com and is in charge Mechanical Turk. She runs the service on a day-to-day basis, consulting with engineers and customers to continually improve the service. Chiarella and Kay Kinton, public relations manager at Amazon, explain a little more about the service, how it works, and what makes it unique.

Sharon Chiarella

NELSON ZHANG: Tell us about the origins of Mechanical Turk. What was its original use? How did it transform into a service for the public?

SHARON CHIARELLA: Mechanical Turk, which launched in 2005, began life as a service that Amazon itself needed. Amazon had millions of web pages that described individual products, and it wanted to weed out duplicate pages. Software could help, but algorithmically eliminating all the duplicates was impossible. We needed an effective way to distribute work to a broad community of workers.

We realized that many companies had similar tasks, tasks that humans could do better than computers, so we decided to make it available as a marketplace where businesses with work, and people looking to do work, could find each other.

So in short, Mechanical Turk is a marketplace for work. Like other Amazon Web Services (S3 and EC2, for instance), Mechanical Turk gives businesses and developers access to on-demand, scalable resources. With S3, it's storage. With Mechanical Turk, the on-demand, scalable resource is human intelligence.

NZ: Is Mechanical Turk profitable to Amazon? How do you achieve profitability and continue to finance innovation on Mechanical Turk with its extremely low margins? Are there intangible benefits?

KAY KINTON: We don't break out financial results for Amazon Mechanical Turk. I can say that we're pleased with the momentum in the business and with feedback from workers and requesters.

NZ: What went into the development of Mechanical Turk? Did you envision its success from the start or was it more of an experiment?

SC: A lot of really talented people, and a lot of energy and customer feedback, has gone into the development of Mechanical Turk. We spend a lot of time with customers—businesses, developers, and workers—to learn how we can continue to evolve the service.

We knew it was a revolutionary idea. We've been pleased with the customer feedback and momentum we've seen. Businesses, from startups to enterprises, are using Mechanical Turk for a broad

> **"Businesses, from startups to enterprises, are using Mechanical Turk for a broad range of use cases, everything from web site content management, to metadata creation, to categorization, to transcription."**

range of use cases, everything from web site content management, to metadata creation, to categorization, to transcription.

NZ: What's the meaning and significance behind the name "Mechanical Turk?"

SC: The name Mechanical Turk is a historical reference to an 18th century chess-playing device. The original Mechanical Turk was powered by a human chess player who was hidden inside the actual device. Typically, a human makes a request of a computer, and the computer does the computation of the task. But Mechanical Turk inverts that. The computer has a task that is easy for a human but extraordinarily hard for the computer. So instead of calling a computer service to perform the function, it calls a human. This is reflected in both our service name and our use of the phrase "artificial, artificial intelligence."

NZ: What kind of role would you say Mechanical Turk and the concept of "artificial artifical intelligence" play in the development of AI as a field?

SC: There are still so many things that humans do better than computers: image recognition, language processing, translation, transcription, OCR, content creation. These tasks require human intelligence *and* Mechanical Turk. We have a lot of AI researchers who use Mechanical Turk to help them create clean data to train their AI algorithms.

NZ: That sounds like a very interesting alternate use. Could you tell me how your AI researchers make use of Mechanical Turk, and some examples of research it has spawned?

SC: AI researchers have used it for a number of different cases. Some have used it to create "gold standards," which they then use to assess how well their AI algorithms complete specific tasks. They also use Mechanical Turk to gather source data. For instance many voice recognition researchers will use Mechanical Turk to gather diverse voice recordings.

We had one requester upload city and state combinations and gather recordings of workers saying these names. This gave the researcher a very broad set of voice clips from workers around the world. The clips were then used to assess the accuracy of their voice recognition algorithm.

We also have had researchers use Mechanical Turk to determine "sentiment" in tweets and forum posts. This data is then used to "train" the AI algorithm which will monitor posts.

These are just a few of the use cases but I think it gives you a sense of the breadth of the application.

NZ: It seems that with the wealth of data that Mechanical Turk can provide for training AI, it's possible that users could reverse that and automate their own work. What is Amazon's policy on the use of automated bots to complete tasks?

SC: Workers are not allowed to use automated bots to submit work. It's against our participation agreement.

NZ: Spam is a real problem on Mechanical Turk. Some have even argued that Mechanical Turk is a market for lemons, where prices are low exactly because requesters expect quality to be low. Does Amazon have any plans to track down spam workers, or will requesters always be responsible for that?

KK: Accuracy is impacted by a number of things including the clarity of the instructions and the skill or qualifications of the worker. The reality is that not every worker is good at every task. We provide a number of tools for requesters (businesses) to improve the accuracy of answers they receive. Many requesters qualify their workers by testing how well they complete specific tasks. Some requesters use agreement by multiple workers as a means to assess accuracy. And other requesters use "gold standards" to test worker's answers. Amazon does evaluate a workers' performance across all requesters and does provide feedback to both workers and requesters regarding quality of work submitted.

NZ: What plans does Amazon have to help connect relevant workers with HITs? Any plans to create sub-Turks where individuals with specific skills like translation, writing expertise, or technical prowess can work on tasks matching their skills? Could Mechanical Turk grow to encompass these new families of requests or task types?

KK: Requesters using Mechanical Turk can test worker's proficiency with their tasks, create their own worker groups, and direct work to these workers. Requesters do this for work ranging from transcription to translation to writing and editorial tasks.

NZ: What is Amazon's perspective on labor law criticism—that many Mechanical Turk workers don't make minimum wage?

KK: A worker's pay really depends on what tasks a worker chooses, how good the work is, and if they are a casual or a full-time worker. If a task pays $0.02 and takes 10 seconds to complete, you will be earning $7.20 per hour. If it takes 6 seconds, then a worker would make $12 per hour.

One of the things workers tell us they like about Amazon Mechanical Turk is that it gives them the flexibility to work as much or as little as they like, and it gives them a wide variety of tasks to choose from. In some cases, our workers have told us that Amazon Mechanical Turk provides much more flexibility than a traditional work environment.

NZ: Could you tell us a bit about your job? What is a day at work like for you?

SC: No two days are alike. I might be speaking at a conference one day and critiquing a new product proposal the next. I spend time everyday with the engineers and product managers who are working on new features.

I also spend time every day with customers. For instance I may have a conference call with a new customer to understand their use case or might meet with a partner who is helping businesses integrate Mechanical Turk into their business applications. I also talk with workers to get their thoughts on features, requesters and HITs in the system. I do HITs to see what work is in the system, and then check out turkernation.com to see what workers are saying about specific HITs and requesters. I really learn a lot from these interactions.

Biography

Nelson Zhang (nelsonz@acm.org) is a senior at Shanghai American School Puxi. After graduating in 2011, he plans to major in electrical engineering and computer science. He previously interned at Amazon China, working as an assistant to the business intelligence engineer.

Michael Bernstein contributed to this article.

HELLO WORLD

Programmatic Access to Wikipedia

by Dmitry Batenkov

Wikipedia is often regarded as today's best-known source of collaborative intelligence. As such, it can also be an excellent subject for research which comes into the domain of "distributed cognition." In this tutorial we will learn how to programmatically access the data behind Wikipedia by using its Web API.

Web API

Web application programming interfaces (APIs) are the standard way of communication in the Web 2.0 environment. There may be several variants, but the most basic one is as follows:

1. The client (for example, the web browser) requests data by sending an HTTP request to the server, optionally passing parameters in the query string.
2. The server returns the result in a well-defined format, usually XML or JSON.

The description of the API methods should somehow be made available to the client.

Accessing MediaWiki

MediaWiki, the wiki engine behind Wikipedia and many other collaborative projects, exposes a public Web API whose entry point varies but in general looks like this: *http://SITE/.../api.php*.

For English Wikipedia, it is *http://en.wikipedia.org/w/api.php*, while for the Polymath project it is *http://michaelnielsen.org/polymath1/api.php*. Pointing your browser to this address will give you a complete documentation of the API. To get a feeling of how it works, let's consider some examples.

Example 1

http://en.wikipedia.org/w/api.php?action=query&list=random&rnnamespace=0

- **action=query**—this is what you will use most of the time unless you want to edit data (when developing bots for example)
- **list=random**—instructs to choose a random page
- **rnnamespace=0**—instructs to select a page in namespace with id=0. More on that later.

The return value from the above call is a data structure which looks like this:

```
<?xml version="1.0"?>
<api>
 <query>
  <random>
   <page id="3997844" ns="0" title="European medieval architecture in North America" />
  </random>
 </query>
</api>
```

The format of the data can be controlled by passing "format" parameter to each API call. Since in this example we didn't supply this parameter, the default value "xmlfm" was used, which means "XML pretty-printed in HTML." You would mostly use this for debugging. In real applications, you will probably specify "xml" or "json."

JSON stands for JavaScript object notation. It is a data interchange format which is both human-readable and can be easily parsed, analogous to XML.

Example 2

Each resource in MediaWiki belongs to a single namespace, for example:

- id=0: normal pages (i.e., content pages)
- id=1: talk pages
- id=14: category pages

To see a complete list of namespaces, issue the following call:

http://en.wikipedia.org/w/api.php?action=query&meta=siteinfo&siprop=namespaces

Using Python API

There are wrappers around the Web API for many scripting languages. In **Listings 1-3**, we demonstrate how to use the Python API. You should install the following prerequisites:

- Python 2.5+
- **python-wikitools** package, which is available at *http://code.google.com/p/python-wikitools/*.

RESOURCES & FURTHER READING

MediaWiki API Main Documentaion
http://www.mediawiki.org/wiki/API

Wikipedia Bots Development
http://en.wikipedia.org/wiki/Wikipedia:Creating_a_bot

MediaWiki Client tools
http://www.mediawiki.org/wiki/API:Client_Code

WikiXRay
http://meta.wikimedia.org/wiki/WikiXRay

Listing 1: Requests to the WebAPI are sent using api.APIRequest objects. Individual pages can be obtained with pagelist.listFromQuery() method which returns a list of page objects.

```
from wikitools import * # wiki, page, pagelist, api, category, ...

site = wiki.Wiki("http://en.wikipedia.org/w/api.php")
params = {'action':'query',
        'list':'random',
        'rnnamespace':0 }
request = api.APIRequest(site, params)

lst = pagelist.listFromQuery(
   site, request.query()['query']['random'])
p = lst[0]
p_url = ''.join( (site.domain, '/wiki/',p.urltitle))

print ''.join(['Title: ',p.title])
print ''.join(['URL: ', p_url])
cats = p.getCategories()
```

Listing 2: A page can be queried for the categories it belongs to.

```
if cats:
   print 'Categories:'
   for ct in pagelist.listFromTitles(site,cats):
      print ''.join(['\t* ',ct.unprefixedtitle])
else:
   print 'No categories'
```

Listing 3: You can easily access the links and backlinks for any page.

```
request2 = api.APIRequest(site, {'action':'query',
                         'titles':p.title,
                         'prop':'extlinks'})
pageinfo = request2.query()['query']['pages'][p.pageid]
if 'extlinks' in pageinfo:
   exts = pageinfo['extlinks']
   print 'External links:'
   for e in exts:
      url = e['*']
      print ''.join(['\t* ', url])
else:
   print 'No external links'

request3 = api.APIRequest(site, {'action':'query',
                         'list':'backlinks',
                         'bltitle':p.title,
                         'blnamespace':0})
blinks = request3.query()['query']['backlinks']
if blinks:
   print 'Backlinks:'
   for bl in blinks:
      print ''.join(['\t* ',bl['title']])
else:
   print 'No backlinks'
```

end

LABZ

FXPAL—An Interdisciplinary Lab
Palo Alto, California

A collegial atmosphere is one of the hallmarks of FXPAL in California.

On the west side of Silicon Valley, nestled in the base of the foothills and slightly to the south of Stanford University sits FXPAL, or FX Palo Alto Laboratory. Despite its small size, or perhaps because of it, this lab has been a nexus for interdisciplinary research, creativity, and technical innovation in the 15 years since its creation.

FXPAL was founded in 1995 as a way to foster innovation and build new business for Fuji Xerox, which is how it got its "FX." Xerox wanted to get a finger on the pulse of what was going on in the booming technology industry. What emerged was a software research lab with a friendly, tight-knit culture, a wide variety of interests, and a wealth of research.

> **"Research interns at FXPAL truly get the opportunity to be a part of the FXPAL family. The company culture is warm and inviting, and there is the distinct sense that people enjoy their work."**

FXPAL is deliberately small, with a staff of only about 25 researchers. Areas of expertise span computer vision, human-computer interaction, information retrieval, mixed and augmented reality, data mining, computer security, and several other fields. This diversity enables a unique environment that few researchers ever have the opportunity to experience: a combination of varied expertise and a place where "everybody knows your name."

In fact, people who work at FXPAL can't help but know everyone else at the lab. It's so small, it only takes a few minutes to walk all the way around the entire building. As research tells us, collocation leads to informal (spontaneous) communication, which can feed creativity and collaboration. Suffice to say, opportunities to work across disciplines on new ideas are numerous.

Despite its tiny staff roster, FXPAL's collaborations aren't confined to what goes on in just one tiny building. The lab partners with multiple external organizations, such as TCHO, a chocolate company located in San Francisco that called upon FXPAL to help it explore ways to leverage technology in its factory, even when nobody is there. FXPAL maintains solid relationships around the San Francisco Bay area, and hosts visitors from both near and far.

THE INTERN EXPERIENCE

Research interns at FXPAL truly get the opportunity to be a part of the FXPAL family. The company culture is warm

5,000

The number of people it would take to label all images on Google in one month using Peek A Boom if they played 24/7.

http://video.google.com/videoplay?docid=-8246463980976635143#

982

The number of volumes an English-language Wikipedia would need if it were printed and bound in book form, with each volume 25cm tall and 5cm thick.

and inviting, and there is the distinct sense that people enjoy their work. Researchers and interns will often throw around new ideas, work through challenging problems, or talk about the winning goal in a sports game from the previous night.

While internships are typically focused around a particular project, FXPAL gives interns the latitude to shape their projects in ways that they find compelling. Nobody is ever simply told what to do. Instead, interns are treated as worthy colleagues with valuable insights. This respect for coworkers of all kinds makes it a pleasure to intern at FXPAL, and leads to another key feature of the environment: look around the intern room and there's not a timecard in sight.

Interns will find very little bureaucratic red tape to deal with at FXPAL. At one point in my internship, I needed some serious equipment to run a mobile phone study. All I had to do was speak up. Before I knew it, I had enough phones to run the study, no questions asked.

The area surrounding FXPAL is home to several technology companies, both large and small, Stanford students, college-town hangouts, and enough open space for running, biking, and hiking. There are plenty of great day-trip destinations nearby, like San Francisco (the local commuter rail Caltrain will get you there in less than an hour), Lake Tahoe, Monterey Bay, and even Yosemite National Park.

GETTING INVOLVED

Last January, FXPAL posted descriptions of all the internships it would have the coming summer at http://palblog.fxpal.com (and perhaps we'll see the same for 2011). Internships are paid and usually last about 12 weeks. Be sure to keep an eye on the blog and apply early.

—Jason Wiese

BACK

The Brain

We've all heard of Moore's Law, which states that the number of transistors placed inexpensively on an integrated circuit doubles approximately every two years. Well, since this issue covers human computation, let's have a look at our own biological hardware. How are we doing compared to a previous model? Keep evolving, folks...

—James Stanier

HOMO ERECTUS	**HOMO SAPIENS**
Time on market 1.8 to 0.3 million years ago	Time on market Roughly 200,000 years ago to present day
CPU size 600–1,150cm3	CPU size 1,150–1,750cm3
Key cultural features Stone hand axes	Key cultural features Language, symbolism
Origin Africa	Origin Africa
Achievements Communication, tool manufacturing, large-game hunting, possible use of fire.	Achievements Electricity, industry, the internet, spaceflight, modern medicine, healthcare, government.
Extinct Yes	Extinct Not yet

EVENT DATES

CONFERENCES & JOURNALS

Annual International Conference on Knowledge Discovery (KD)
Phuket Beach Resort
Phuket, Thailand
December 6–7, 2010
520 SGD
http://www.kdiscovery.org/index.html

8th International Conference on Service Oriented Computing
Hilton Financial District
San Francisco, California
December 7–10, 2010
Cost: $300–$400 (student)
http://www.icsoc.org/

ACM Transactions on Internet Technology
Context-Aware Web Services for the Future Internet
Submission deadline: December 15, 2010
Q3 2011 issue
http://toit.acm.org/

International Conference on e-Business, Management and Economics (ICEME)
Hong Kong, China
December 28–30, 2010
Cost: $350 US (student)
http://www.iceme.org/

IEEE Internet Computing
Smart Grids
Submission deadline: January 4, 2011
September–October 2011 issue
http://www.computer.org/portal/web/computingnow/iccfp5

Foundation of Genetic Algorithms XI (FOGA)
Hotel Gasthof Hirschen Schwarzenberg
Schwarzenberg, Austria
January 5–9, 2011
Cost: $900–$1,000 (student)
http://www.sigevo.org/foga-2011/index.html

3rd International Conference on Next Generation Networks and Services
Hammamet, Tunisia
Submission deadline: January 17, 2011
Conference: May 20–22, 2011
http://ngns2011.regim.org/

International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC)
Rome, Italy
January 26–29, 2011
http://www.biostec.org/

38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)
Austin, Texas, USA
January 26–28, 2011
http://www.cse.psu.edu/popl/11/

19th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)
University of West Bohemia, Campus-Bory
Plzen, Czech Republic
January 31–February 3, 2011
Cost: €280–€340
http://wscg.zcu.cz/WSCG2011/wscg2011.htm

4th ACM International Conference on Web Search and Data Mining (WSDM)
Sheraton Hotel and Towers
Hong Kong, China
February 9–12, 2011
http://www.wsdm2011.org/

International Conference on Intelligent Use Interfaces (IUI)
Sheraton Palo Alto
Palo Alto, California
February 13–16, 2011
http://iuiconf.org/

11th IASTED International Conference on Artificial Intelligence and Applications
Innsbruck, Austria
February 14–16, 2011
http://www.iasted.org/conferences/home-717.html

International Conference on Fuzzy Systems and Neural Computing
Hong Kong, China
February 20–21 2011
$400 US (student), $200 (listener)
http://www.iita-association.org/fsnc2011/index.htm

5th International Conference on Ubiquitous Information Management and Communication (ICUIMC)
Sungkyunkwan University
Seoul, Korea
February 21–23
http://www.icuimc.org/

4th International Conference on Advances in Computer-Human Interactions (ACHI)

FEATURED EVENT

ICDM 2010—The 10th IEEE Conference on Data Mining
Sydney, Australia
December 14–17, 2010

Do you think "miners" only produce ore? Well, computer miners love digging as well. They may not trudge through deep, dirty holes in the earth, but sitting in front of their PCs, they implement powerful tools to process huge amounts of data in a really smart way. The best of these miners can be found every year in the IEEE International Conference on Data Mining (ICDM).

Statistics, machine learning, pattern recognition, databases and data warehousing, data visualization, knowledge-based systems, and high performance computing are some of the areas whose experts join this conference. Inside the conference's venue you will find them talking about all aspects of data mining, including algorithms, software and systems, and applications.

Of course, the technical program would not stand alone in such a conference. About 20 workshops, many tutorials, a colloquium, and a contest will complement IDCM where three of the best academics in the field will talk as keynote speakers. The organization committee has been able to offer some travel grants for students and provide some awards for participants.
http://datamining.it.uts.edu.au/icdm10/

—Vaggelis Giannikas

Gosier, Guadeloupe, France
February 23–28 2011
http://www.iaria.org/conferences2011/ACHI11.html

5th International Conference on Digital Society (ICDS)
Gosier, Guadeloupe, France
February 23–28, 2011
http://www.iaria.org/conferences2011/ICDS11.html

IEEE Computer Graphics and Applications
Digital-Content Authoring
Submission deadline: February 28, 2011
November–December 2011 issue
http://www.computer.org/portal/web/computingnow/cgacfp6

12th Workshop on Mobile Computing Systems and Applications (HotMobile)
Pointe Hilton, Squaw Peak Resort
Phoenix, Arizona
March 1–2, 2011
http://www.hotmobile.org/2011/Home.html

International Conference on Systems, Man, and Cybernetics (SMCS)
Hilton Anchorage
Anchorage, Alaska
Submission deadline: March 5, 2011
Conference: October 9 2011
http://www.ieee.org/conferences_events/conferences/conferencedetails/index.html?Conf_ID=15399

CONTESTS AND EVENTS

Score 2011 in Hawaii!
The Student Contest on Software Engineering (SCORE) is a worldwide competition for student teams at the undergraduate and master's level. Student teams participating in the contest are able to choose from a number of project topics proposed by the SCORE Program Committee, which cover diverse application fields and types.
A representative of each finalist team will receive financial support to partially defray the cost of travel and participation in the SCORE 2011 finals at ICSE 2011 in Hawaii. Evaluation will be based on the quality of all aspects of the software engineering process followed, as well as the outcome. The SCORE 2011 Contest will run from February 2010 to January 2011, with team registration ending in November 2010 and project submission starting in February 2010.
http://score-contest.org/2011/

ACM Student Research Competition at Design Automation Conference
Sponsored by Microsoft Research, The ACM Special Interest Group on Design Automation is organizing such an event in conjunction with the Design Automation Conference. Authors of accepted submissions will get travel grants from ACM/Microsoft to attend the event at DAC. The event consists of several rounds, as described at and, where you can also find more details on student eligibility and timeline.
http://www.acm.org/src

SIGDA University Booth at DAC
The SIGDA/DAC University Booth has been providing an opportunity for the university community to demonstrate EDA tools, design projects, and instructional materials at the Design Automation Conference. The University Booth provides demonstration space, displays, and a wireless Internet connection for participating universities. The SIGDA/DAC University Booth also provides modest travel grants to the participating students to cover transportation and lodging expenses.

GRANTS, SCHOLARSHIPS & FELLOWSHIPS

Intel Science Talent Search
Deadline: November 17, 2010

Eligibility: High school seniors from around the United States. The competition covers independent science, math or engineering research.

Benefits: $1.25 million in awards each year. Each of the 300 students named a Semifinalist will receive a $1,000 award (each school will receive the same award). The main prizes range from $7,500 to $100,000.

Explanation: Since 1942, it is America's most prestigious pre-college science competition. It is a world-renowned competition offered by the Society for Science & the Public (SSP). It provides original research that is reviewed by a national jury of professional scientists. Alumni have made extraordinary contributions and have received recognition worldwide, including some Nobel Prizes and National Medals.
http://www.societyforscience.org/sts

AACE International Competitive Scholarships
Deadline: February 15, 2011

Eligibility: Applicants must be enrolled full-time students in a program related to cost management. It includes agricultural, architectural, building, business, chemical, civil, industrial, manufacturing, mining and electrical engineering. Selection criteria are: academic performance (35%), extracurricular activities (35%) and an essay (30%) explaining why cost management will be important in your career goals.

Benefits: $2,000-$8,000 for one year.

Explanation: AACE International is an organization dedicated to advancing the study of cost engineering, and cost management through the integrative process of total cost management. To promote this objective, AACE awards academic scholarships to full-time students pursuing a related degree. The cost engineering profession uses the process of total cost management to identify and define those areas of business in which the discipline of cost engineering and cost management principles may be applied to plan and control resources, assets, costs, profitability, and risk.
http://www.aacei.org/education/scholarship/

AFCEA Distance Learning/Online Scholarship
Deadline: June 1, 2011

Eligibility: Applicants must be U.S. citizens currently enrolled in a full-time distance-learning or online degree program, pursuing either a Bachelor of Science or Master of Science degree.

Benefits: $2,000 for one year.

Explanation: The AFCEA Educational Foundation is a non-profit organization dedicated to providing educational incentives and assistance for people

EVENT DATES

engaged in information management, communications and intelligence efforts and to fostering excellence in education, particularly in the "hard science" disciplines related to Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance. The AFCEA Distance Learning/Online Scholarship focuses on electrical, chemical, systems or aerospace engineering; mathematics; physics; science or mathematics education; technology management; and computer science.
http://www.afcea.org/education/scholarships/undergraduate/pub1.asp

---

AFCEA Scholarship for Working Professionals
Deadline: September 1, 2011

Eligibility: Applicants must be U.S. citizens enrolled in a degree program at an accredited college or university, and employed in one of the science and technology disciplines directly related to the mission of AFCEA. Undergraduate and graduate applications will be accepted under certain conditions. The scholarship is intended for students attending school on a part-time basis.

Benefits: $2,000 for one year.

Explanation: The AFCEA Educational Foundation is a non-profit organization dedicated to providing educational incentives and assistance for people engaged in information management, communications and intelligence efforts and fostering excellence in education particularly in the "hard science" disciplines related to Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance. The AFCEA Scholarship for Working Professionals focuses on electrical, chemical, systems or aerospace engineering; mathematics; physics; technical management; computer information systems; computer science or related fields.
http://www.afcea.org/education/scholarships/workingstudents/ws1.asp

---

AFCEA VADM Samuel L. Gravely, Jr., USN (Ret.) Memorial Scholarship
Deadline: September 1, 2011

Eligibility: Applicants must be U.S. citizens pursuing an undergraduate degree in an eligible major at an accredited Historically Black College or University institution. The first scholarships will be awarded in 2010. Students enrolled in distance learning or online degree programs are eligible.

Benefits: At least two awards of $5,000 for one year.

Explanation: These scholarships honor the memory of Vice Admiral, USN (Ret.), Samuel L. Gravely, Jr., the Educational Foundation's first executive director. He was also the first African-American to achieve flag rank and eventually Vice Admiral. The AFCEA VADM Samuel L. Gravely, Jr., USN (Ret.) Memorial Scholarship focuses on electrical, aerospace, systems or computer engineering; computer engineering technology; computer information systems; information systems management; computer science; physics; or mathematics. AMERICAN SYSTEMS Corporation supports the scholarship.
http://www.afcea.org/education/scholarships/undergraduate/Gravely.asp

---

AAUW American Fellowships
Deadline: November 15, 2010

Eligibility: Applicants must be U.S. citizens or permanent residents. Fellowships are for women working on doctoral dissertations, or pursuing postdoctoral research who have completed their dissertation by the application deadline.

Benefits: $20,000–$30,000 for one year.

Explanation: The American Association of University Women is a network for more than 100,000 members and donors. AAUW advances equity for women through advocacy, education and research, breaking educational and economic barriers so that all women have a fair chance. The AAUE American Fellowships program is the oldest and largest of the AAUW's fellowship and grant programs.
http://www.aauw.org/learn/fellows_directory/american.cfm

FEATURED FELLOWSHIP

American Association of University Women (AAUW) International Fellowships

Deadline: December 1, 2010
Fellowship year: July 1, 2011–June 30, 2012
Benefits:
Master's/Professional Fellowship: $18,000
Doctorate Fellowship: $20,000
Postdoctoral Fellowship: $30,000

AAUW International Fellowships are for women pursuing either graduate or postgraduate studies at accredited institutions who are not citizens or residents of the United States. The majority return to their home countries to become leaders in their fields.

The American Association of University Women is a network for more than 100,000 members and donors. AAUW advances equity for women through advocacy, education and research, breaking educational and economic barriers so that all women have a fair chance. The AAUW International Fellowships program dates from 1917 and was originally designed to provide Latin American women with opportunities for graduate and postgraduate study in the United States. Forty-one fellowships are typically awarded.
http://www.aauw.org/learn/fellows_directory/international.cfm

The organization also has an American Fellowship for U.S. permanent residents and citizens (see *http://www.aauw.org/learn/fellowships_grants/american.cfm*).

HUMAN COMPUTATION GROUPS

ACM Special Interest Group on Algorithms and Computation
SIGACT is an international organization that fosters and promotes the discovery and dissemination of high quality research in theoretical computer science (TCS), the formal analysis of efficient computation and computational processes.
http://www.sigact.org/

ACM's Special Interest Group on Artificial Intelligence
SIGART focuses on the study of intelligence and its realization in computer systems. SIGART's mission is to promote and support AI-related conferences. Members receive reduced registration rates to all affiliated conferences. Members also receive proceedings from the major SIGART-sponsored conferences.
http://www.sigart.org/

ACM Special Interest Group on Computer-Human Interaction
SIGCHI is the premier international society for professionals, academics and students who are interested in human-technology and human-computer interaction (HCI). They provide a forum for the discussion of all aspects of HCI through their conferences, publications, web sites, email discussion groups and other services.
http://www.sigchi.org/

ACM Special Interest Group on Computers and Society
SIGCAS addresses the social and ethical consequences of widespread computer usage. SIGCAS' main goals are to raise awareness about the impact that technology has on society, and to support and advance the efforts of those who are involved in this important work.
http://www.sigcas.org/

ACM Special Interest Group on Genetic and Evolutionary Computation
SIGEVO operates an annual Genetic and Evolutionary Computation Conference (GECCO), which combines the formerly held International Conference on Genetic Algorithms and Genetic Programming Conferences. They also support and periodically operate other specialized conferences and events, provide support for student participation in the activities of the organization and educational activities, and promote public information about the field of genetic and evolutionary computation.
http://www.sigevo.org/

ACM Special Interest Group on Knowledge Discovery in Data Mining
SIGKDD's primary mission is to provide the premier forum for advancement, education, and adoption of the "science" of knowledge discovery and data mining from all types of data stored in computers and networks of computers. SIGKDD promotes basic research and development in KDD, adoption of "standards" in the market in terms of terminology, evaluation, methodology, and interdisciplinary education among KDD researchers, practitioners, and users.
http://www.sigkdd.org/

ACM Special Interest Group on Software Engineering
SIGSOFT seeks to improve our ability to engineer software by stimulating interaction among practitioners, researchers, and educators; by fostering the professional development of software engineers; and by representing software engineers to professional, legal, and political entities.
http://www.sigsoft.org/

HUMAN COMPUTATION RESOURCES

Book: Here Comes Everybody: The Power of Organizing Without Organizations
Clay Shirky, Penguin Press HC (2008)
Clay Shirky contextualizes the digital networking age with philosophical, sociological, economic, and statistical theories and points to its major successes and failures.

Book: The Wisdom of Crowds
James Surowiecki, Anchor (2005)
James Surowiecki explores problems involving cognition, coordination, and cooperation. It offers a great introduction to game theory and covers a range of problems, including driving in traffic, competing on TV game shows, maximizing stock market performance, voting for political candidates, navigating busy sidewalks, tracking SARS, and designing Internet search engines like Google.

Book: Programming Collective Intelligence
Toby Segaran, O'Reilly Media (2007)
Toby Segaran explains how to build Web 2.0 applications to mine the enormous amount of data created by people on the Internet. It shows the world of machine learning and statistics, and explains how to draw conclusions about user experience, marketing, personal tastes, and human behavior.

Handy URLS
http://aws.amazon.com/mturk/
http://behind-the-enemy-lines.blogspot.com/
http://crowdflower.com/
http:// www.fxpal.com
http://www.gwap.com/
http://vonahn.blogspot.com/
http://www.turkernation.com/

GRADUATE PROGRAMS

Carnegie Mellon University
Software Engineering Masters Programs
http://mse.isri.cmu.edu/software-engineering

Stanford University|
School of Engineering
http://soe.stanford.edu

University of Canterbury
College of Engineering
http://www.cosc.canterbury.ac.nz

University of Melbourne
Melbourne School of Engineering
http://csse.unimelb.edu.au

University of Minnesota
Master of Science in Software Engineering
http://www.msse.umn.edu

University of Oviedo
School of Computer Science
http://www.informatica.uniovi.es/home/en

University of Oxford
Software Engineering Programme
http://www.softeng.ox.ac.uk

BEMUSEMENT

Puzzle No Tipping

This first puzzle comes from the field of physics, Newtonian mechanics in fact. As Archimedes famously observed and as every child on a seesaw reconfirms, if you put a heavy object far out on a lever arm, it will exert a twisting influence around any support. That twisting influence is called "torque" and is equal to the weight times the distance (the angle also comes in, but that does not concern us here). If the object is to the left of the fulcrum, the direction of the torque is counterclockwise, and vice versa. To compute the torque around a support simply sum all the torques of the individual objects.

For example, if a 10 meter board weighs 2 kilograms and its center of mass is at its middle, and we put a fulcrum at 3 meters, then the board will twist clockwise with a torque of 3x2=6 kilogram-meters. If we put a 5-kilogram weight at the very left end of the board, it will cause a counterclockwise torque of 5x3=15 kilogram-meters. The net torque will be counterclockwise 15−6=9 kilogram-meters.

Now to the puzzle: You are presented with a straight, evenly weighted board 20 meters long and weighing 3 kilograms. The middle of the board (10 meters from the left end) is the center of mass. We call that position 0. So possible positions range from −10 to +10. The board is supported at −1.5 and +1.5 by two equal supports both 10 meters from a flat floor.

On the board are 15 packages at various positions. Remove the packages one at a time in such a way that the board rests on both supports without tipping. The board will tip if the net torque around the left support (due to the packages and the board) was counterclockwise or the net torque around the right support was clockwise.

Suppose there are 6 packages at positions −8, −4, −3, 2, 5, 8 and having weights 4, 10, 10, 4, 7, 8. Before reading on, try to figure out how to remove the packages so that the board never tips.

Now for the real problem: There are 15 packages, some are at the same positions relative to the center but side-by-side across the board (so you can remove them in any order; see the table).

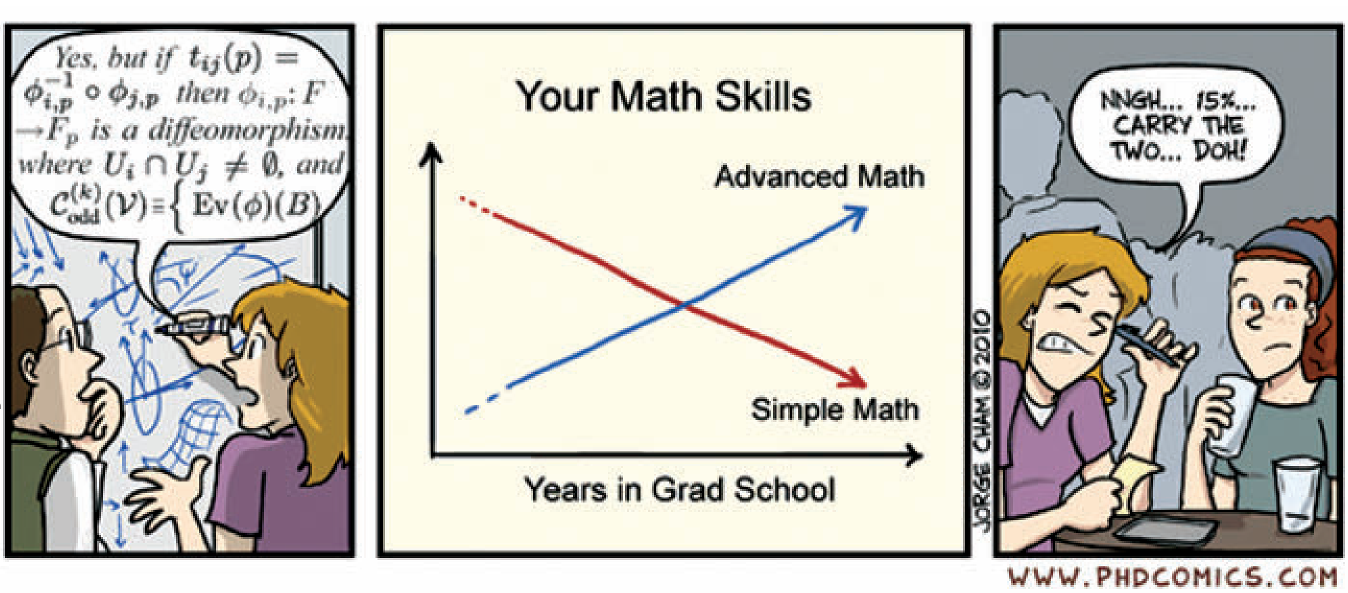
Package	Position	Weight
1	-8	4
2	-6	8
3	-6	1
4	-4	5
5	-3	10
6	-3	2
7	-2	2
8	1	10
9	2	9
10	2	5
11	3	3
12	5	9
13	5	1
14	8	5
15	8	10

Find an order of removing packages such that the board never tips. Tyler Neylon has developed a two-person No Tipping game which you can find at: http://cs.nyu.edu/courses/fall05/G22.2965-001/notipping.d/index.html

—*Dennis E. Shasha*

Find the solution at: http://xrds.acm.org/bemusement-solutions/2010.cfm

A Mathematical Paradox



PhD Comics ©Jorge Cham

SUBMIT A PUZZLE

Can you can do better? Bemusements would like your puzzles and mathematical games (but not Sudoku). Contact **xrds@acm.org** to submit yours!

STUDENT MEMBERSHIP APPLICATION

Join ACM online: www.acm.org/joinacm

CODE: CRSRDS

Name ______ *Please print clearly*

Address ______

City ______ State/Province ______ Postal code/Zip ______

Country ______ E-mail address ______

Area code & Daytime phone ______ Mobile Phone ______ Member number, *if applicable* ______

MEMBERSHIP BENEFITS AND OPTIONS

- Print & electronic subscription to *XRDS: The ACM Magazine for Students*
- Electronic subscription to *Communications of the ACM* magazine
- Free software and courseware through the ACM Student Academic Initiative
- 3,500 online courses in multiple languages, 1,100 virtual labs and 500 online books
- ACM's *CareerNews* (twice monthly)
- ACM e-news digest *TechNews* (tri-weekly)
- Free e-mentoring service from MentorNet
- ACM online newsletter *MemberNet* (twice monthly)
- *Student Quick Takes*, ACM student e-newsletter (quarterly)
- ACM's Online Guide to Computing Literature
- Free "acm.org" email forwarding address plus filtering through Postini
- Option to subscribe to the full ACM Digital Library
- Discounts on ACM publications and conferences, valuable products and services, and more

PLEASE CHOOSE ONE:

❑ **Student Membership: $19 (USD)**
❑ **Student Membership PLUS Digital Library: $42 (USD)**
❑ **Student Membership PLUS Print *CACM* Magazine: $42 (USD)**
❑ **Student Membership w/Digital Library PLUS Print *CACM* Magazine: $62 (USD)**

PUBLICATIONS

Please check one

Check the appropriate box and calculate amount due.	**Issues per year**	**Code**	**Member Rate**	**Member Rate PLUS Air***
• ACM Inroads	4	247	$13 ❐	$65 ❐
• *acmqueue* website www.acmqueue.com	N/A	N/A	N/A	N/A
• Computers in Entertainment (online only)	4	247	$40 ❐	N/A
Computing Reviews	12	104	$52 ❐	$85 ❐
• Computing Surveys	4	103	$33 ❐	$59 ❐
• XRDS: Crossroads, The ACM Magazine for Students (included with student membership)	4	XRoads	N/A	N/A
• interactions (included in SIGCHI membership)	6	123	$19 ❐	$48 ❐
Int'l Journal of Network Management (Wiley)	6	136	$90 ❐	$116 ❐
• Int'l Journal on Very Large Databases	4	148	$81 ❐	$107 ❐
• Journal of Experimental Algorithmics (online only)	12	129	$29 ❐	N/A
• Journal of Personal and Ubiquitous Computing	6	144	$63 ❐	$89 ❐
• Journal of the ACM	6	102	$52 ❐	$104 ❐
• Journal on Computing and Cultural Heritage	4	173	$46 ❐	$63 ❐
• Journal on Data and Information Quality	4	171	$46 ❐	$65 ❐
• Journal on Emerging Technologies in Computing Systems	4	154	$39 ❐	$56 ❐
• Linux Journal (SSC)	12	137	$29 ❐	$58 ❐
• Mobile Networks & Applications	4	130	$70 ❐	$95 ❐
• Wireless Networks	4	125	$70 ❐	$95 ❐
Transactions on:				
• Accessible Computing	4	174	$46 ❐	$64 ❐
• Algorithms	4	151	$49 ❐	$66 ❐
• Applied Perception	4	145	$40 ❐	$57 ❐
• Architecture & Code Optimization	4	146	$40 ❐	$57 ❐
• Asian Language Information Processing	4	138	$36 ❐	$53 ❐
• Autonomous and Adaptive Systems	4	158	$38 ❐	$55 ❐
• Computational Biology and Bio Informatics	4	149	$18 ❐	$63 ❐
• Computer-Human Interaction	4	119	$40 ❐	$59 ❐
• Computational Logic	4	135	$41 ❐	$60 ❐
• Computation Theory		176	$46 ❐	$72 ❐
• Computer Systems	4	114	$44 ❐	$63 ❐
• Computing Education (online only)		277	$38 ❐	N/A
• Database Systems	4	109	$43 ❐	$62 ❐
• Design Automation of Electronic Systems	4	128	$40 ❐	$59 ❐
• Embedded Computing Systems	4	142	$41 ❐	$58 ❐
• Graphics	4	112	$48 ❐	$67 ❐
• Information & System Security	4	134	$41 ❐	$58 ❐
• Information Systems	4	113	$44 ❐	$63 ❐
• Internet Technology	4	140	$39 ❐	$56 ❐
• Knowledge Discovery From Data	4	170	$47 ❐	$64 ❐
• Mathematical Software	4	108	$44 ❐	$63 ❐
• Modeling and Computer Simulation	4	116	$48 ❐	$67 ❐
• Multimedia Computing, Communications, and Applications	4	156	$39 ❐	$56 ❐
• Networking	6	118	$28 ❐	$76 ❐
• Programming Languages & Systems	6	110	$56 ❐	$82 ❐
• Reconfigurable Technology & Systems	4	172	$46 ❐	$64 ❐
• Sensor Networks	4	155	$39 ❐	$56 ❐
• Software Engineering and Methodology	4	115	$40 ❐	$59 ❐
• Speech and Language Processing (online only)	4	153	$39 ❐	N/A
• Storage	4	157	$39 ❐	$56 ❐
• Web	4	159	$38 ❐	$55 ❐

marked • are available in the ACM Digital Library
**Check here to have publications delivered via Expedited Air Service. For residents outside North America only.*

PUBLICATION SUBTOTAL: ______

INSTRUCTIONS

Carefully complete this application and return with payment by mail or fax to ACM. You must be a **full-time** student to qualify for student rates.

CONTACT ACM

phone: 800-342-6626 (US & Canada)
+1-212-626-0500 (Global)

hours: 8:30am–4:30pm US Eastern Time

fax: +1-212-944-1318

email: acmhelp@acm.org

mail: Association for Computing Machinery, Inc.
General Post Office
P.O. Box 30777
New York, NY 10087-0777

For immediate processing, FAX this application to +1-212-944-1318.

PAYMENT INFORMATION

Payment must accompany application

Member dues ($19, $42, or $62) $ ______

To have *Communications of the ACM* sent to you via Expedited Air Service, add $52 here (for residents outside of North America only). $ ______

Publications $ ______

Total amount due $ ______

Check or money order (make payable to ACM, Inc. in U.S. dollars or equivalent in foreign currency)

❑ **Visa/Mastercard** ❑ **American Express**

Card number ______ Exp. date ______

Signature ______

Member dues, subscriptions, and optional contributions are tax deductible under certain circumstances. Please consult with your tax advisor.

EDUCATION

Name of School ______

Please check one: ❐ **High School** (Pre-college, Secondary School) **College:** ❐ Freshman/1st yr. ❐ Sophomore/2nd yr. ❐ Junior/3rd yr. ❐ Senior/4th yr. **Graduate Student:** ❐ Masters Program ❐ Doctorate Program ❐ Postdoctoral Program ❐ **Non-Traditional Student**

Major ______ Expected mo./yr. of grad. ______

Age Range: ❐ 17 & under ❐ 18-21 ❐ 22-25 ❐ 26-30 ❐ 31-35 ❐ 36-40 ❐ 41-45 ❐ 46-50 ❐ 51-55 ❐ 56-59 ❐ 60+

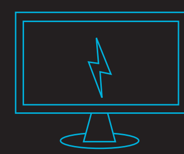
Do you belong to an ACM Student Chapter? ❐ Yes ❐ No

I attest that the information given is correct and that I will abide by the ACM Code of Ethics. I understand that my membership is non transferable.

Signature ______

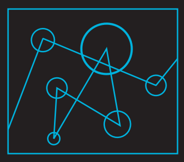
(NOT) YOUR ORDINARY DAY

Bike to work.
Dodge hybrids.

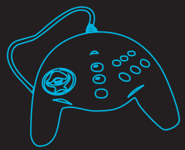


Integrate
hack day project
into the product.

Sushi and
Speed Hearts on
the balcony.



Implement new
force-directed
graph layout
algorithm.



After-dinner
Halo "meeting."

Hang cape.
The world is safe
for another day.

We're actively recruiting superheroes. We invite you to apply today.

Palantir

www.palantir.com/careers