COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

05/2017 VOL.60 NO.05

Who Owns the Social Web?

Safari File Edit View History Bookmarks Window Help
IMAGE REMOVED FOR COPYRIGHT INFRINGEMENT

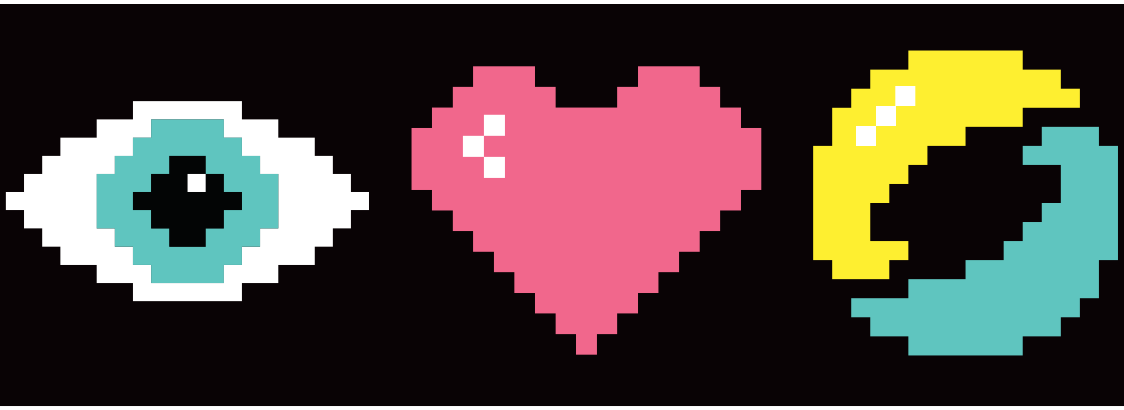
Contest Theory

Making Chips Smarter

Toward a Ban on Lethal Autonomous Weapons

Cyber Insecurity and Cyber Libertarianism

Association for Computing Machinery



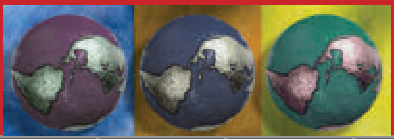
THE ACM A. M. TURING AWARD

by the community ◆ from the community ◆ for the community

ACM and Google congratulate

SIR TIM BERNERS-LEE

For inventing the World Wide Web, the first Web browser, and the fundamental protocols and algorithms allowing the Web to scale.



"The Web has radically changed the way we share information and is a key factor for global economic growth and opportunity" said Andrei Broder, Google Distinguished Scientist. "The idea of a web of knowledge originated in a brilliant 1945 essay by Vannevar Bush. Through 1989, several pieces of the puzzle came together: hypertext, the Internet, personal computing. But the explosive growth of the Web started when Tim Berners-Lee proposed a unified user-interface to all types of information supported by a new transport protocol. This was a significant inflection point, setting the stage for everyone in the world, from high schoolers to corporations, to build their Web presences and collectively create the wonderful World Wide Web."

Andrei Broder
Google Distinguished Scientist
Google Inc.

Google™

For more information see http://research.google.com/

Financial support for the ACM A. M. Turing Award is provided by Google Inc.

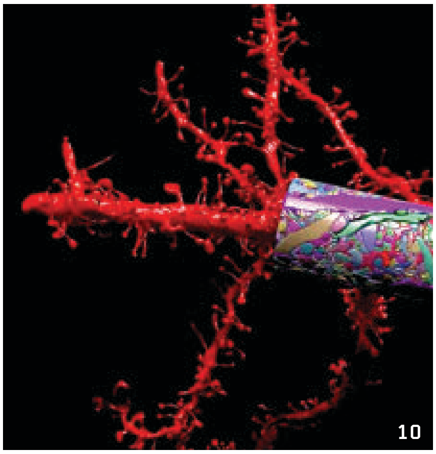
COMMUNICATIONS OF THE ACM

Departments

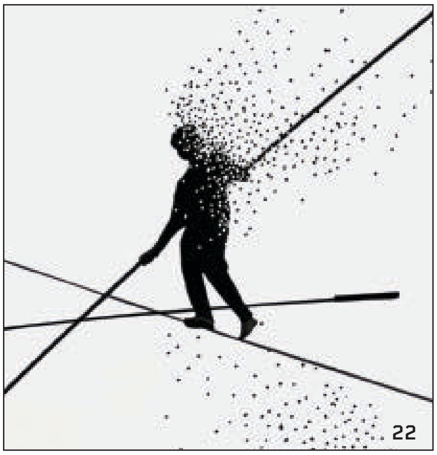
Last Byte

Association for Computing Machinery
Advancing Computing as a Science & Profession

News



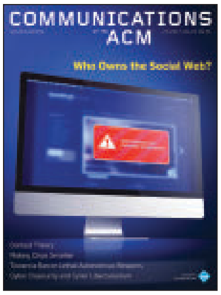
Viewpoints



LEFT: IMAGE COURTESY OF ARGONNE NATIONAL LABORATORY; RIGHT: IMAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

Practice

Articles' development led by acmqueue
queue.acm.org

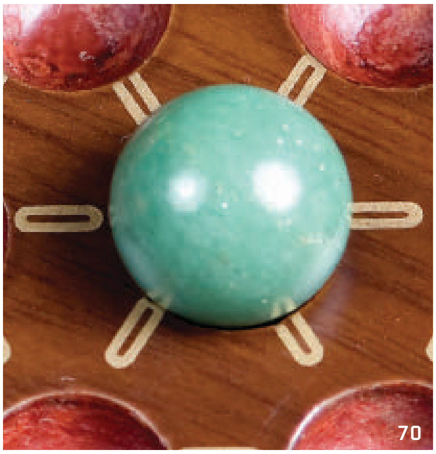


About the Cover:
This month's cover story is a study of attitudes. Catherine Marshall and Frank Shipman (p. 52) share how social media users view who owns what online. Their studies trace how the traditional concepts of "ownership" have drifted afar. Cover illustration by Justin Metz.

Contributed Articles

Watch the authors discuss their work in this exclusive *Communications* video.
http://cacm.acm.org/videos/who-owns-the-social-web

Review Articles



Watch the author discuss his work in this exclusive *Communications* video.
http://cacm.acm.org/videos/contest-theory

Research Highlights

LEFT TO RIGHT: IMAGES BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK, SUAT GURSOZLU, THOMAS M. PERKINS

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

Communications of the ACM is the leading monthly print and online magazine for the computing and information technology fields. *Communications* is recognized as the most trusted and knowledgeable source of industry information for today's computing professional. *Communications* brings its readership in-depth coverage of emerging areas of computer science, new trends in information technology, and practical applications. Industry leaders use *Communications* as a platform to present and debate various technology implications, public policies, engineering challenges, and market trends. The prestige and unmatched reputation that *Communications of the ACM* enjoys today is built upon a 50-year commitment to high-quality editorial content and a steadfast dedication to advancing the arts, sciences, and applications of information technology.

ACM, the world's largest educational and scientific computing society, delivers resources that advance computing as a science and profession. ACM provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.

STAFF
DIRECTOR OF PUBLICATIONS
Scott E. Delman
cacm-publisher@cacm.acm.org

Executive Editor
Diane Crawford
Managing Editor
Thomas E. Lambert
Senior Editor
Andrew Rosenbloom
Senior Editor/News
Larry Fisher
Web Editor
David Roman
Rights and Permissions
Deborah Cotton
Editorial Assistant
Jade Morris

Art Director
Andrij Borys
Associate Art Director
Margaret Gray
Assistant Art Director
Mia Angelica Balaquiot
Designer
Iwona Usakiewicz
Advertising Sales Account Manager
Ilia Rodriguez

Columnists
David Anderson; Phillip G. Armour; Michael Cusumano; Peter J. Denning; Mark Guzdial; Thomas Haigh; Leah Hoffmann; Mari Sako; Pamela Samuelson; Marshall Van Alstyne

CONTACT POINTS
Copyright permission
permissions@hq.acm.org
Calendar items
calendar@cacm.acm.org
Change of address
acmhelp@acm.org
Letters to the Editor
letters@cacm.acm.org

WEBSITE
http://cacm.acm.org

AUTHOR GUIDELINES
http://cacm.acm.org/

ACM ADVERTISING DEPARTMENT
2 Penn Plaza, Suite 701, New York, NY
10121-0701
T (212) 626-0686
F (212) 869-0481

Advertising Sales Account Manager
Ilia Rodriguez
ilia.rodriguez@hq.acm.org

For display, corporate/brand advertising:
Craig Pitcher
pitcherc@acm.org T (408) 778-0300

Media Kit acmmediasales@acm.org

Association for Computing Machinery (ACM)
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA
T (212) 869-7440; F (212) 869-0481

Subscriptions
An annual subscription cost is included in ACM member dues of $99 ($40 of which is allocated to a subscription to *Communications*); for students, cost is included in $42 dues ($20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is $269.

ACM Media Advertising Policy
Communications of the ACM and other ACM Media publications accept advertising in both print and electronic formats. All advertising in ACM Media publications is at the discretion of ACM and is intended to provide financial support for the various activities and services for ACM members. Current advertising rates can be found by visiting http://www.acm-media.org or by contacting ACM Media Sales at (212) 626-0686.

Single Copies
Single copies of *Communications of the ACM* are available for purchase. Please contact acmhelp@acm.org.

COMMUNICATIONS OF THE ACM
(ISSN 0001-0782) is published monthly by ACM Media, 2 Penn Plaza, Suite 701, New York, NY 10121-0701. Periodicals postage paid at New York, NY 10001, and other mailing offices.

POSTMASTER
Please send address changes to
Communications of the ACM
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA

Printed in the U.S.A.

Association for
Computing Machinery

PLEASE RECYCLE THIS MAGAZINE

DOI:10.1145/3073731

Moshe Y. Vardi

Cyber Insecurity and Cyber Libertarianism

One can get a good picture of what is "hot" in technology by attending a Tech Summit. Such events are now held regularly in places trying to compete with Silicon Valley. I attended such a summit a few weeks ago. So what's hot? FinTech (financial technology), MedTech (medical technology), IoT (Internet of Things), and autonomous cars are all hot. These areas attract a high level of venture capital, and one can expect them to grow and reshape the financial, medical, and transportation industries. Underlying these technologies is, of course, the Internet—our "network of insecurity"—so we can expect cyber insecurity to spread across more and more aspects of our lives.

Cyber insecurity seems to be the normal state of affairs these days. In June 2015, the U.S. Office of Personnel Management announced it had been the target of a data breach targeting the records of as many as 18 million people. In late 2016, we learned about two data breaches at Yahoo! Inc., which compromised over one billion accounts. Lastly, during 2016, close to 20,000 email messages from the U.S. Democratic National Committee were leaked via WikiLeaks. U.S. intelligence agencies argued that the Russian government directed the breaches in an attempt to interfere with the U.S. election process. Furthermore, cyber insecurity goes way beyond data breaches. In October 2016, for example, emergency centers in at least 12 U.S. states had been hit by a deluge of fake emergency calls. What cyber disaster is going to happen next?

So here we are, 70 years into the computer age and after three ACM Turing Awards in the area of cryptography (but none in cybersecurity), and we still do not seem to know how to build secure information systems. This state of affairs was bemoaned in 2005 by then ACM President David Patterson, who argued (https://goo.gl/9QbuZc), "We must protect the security and privacy of computer and communication users from criminals and terrorists while preventing the Orwellian vision of Big Brother." Yet here we are, over a decade later, and Patterson's passionate appeal is as relevant as ever! That is not to say we have not made significant progress in the development of security-enhancing techniques, but we have not really succeeded in making information-technology infrastructure more secure. As information technology permeates more and more aspects of our lives, the stakes are getting higher and higher. The risk is no longer merely about compromised privacy. We must worry now about the integrity of vital infrastructure components, including the electrical-power grid, the telecommunication system, the financial system, and the transportation system. And yet, our community marches forward with no special sense of urgency.

The basic problem, I believe, is that security never gets a high-enough priority. We build a computing system for certain functionality, and functionality sells. Then we discover security vulnerabilities and fix them, and security of the system does improve. Microsoft Windows 10 is much, much better security-wise than Windows XP. The question is whether we are eliminating old vulnerabilities faster than we are creating new ones. Judging by the number of publicized security breaches and attacks, the answer to that question seems to be negative.

This raises some very fundamental questions about our field. Are we investing enough in cybersecurity research? Has the research yielded solid scientific foundations as well as useful solutions? Has industry failed to adopt these solutions due to cost/benefit? More fundamentally, how do we change the trajectory in a fundamental way, so the cybersecurity derivative goes from being negative to being positive?

We can draw an analogy to car safety. Over the past 100 years, the amount of vehicle miles traveled has been steadily increasing, but fatalities with respect to vehicle miles traveled have been decreasing. Car safety has been increasing mostly due to government regulation. For example, the U.S. Congress established the National Transportation Safety Board in 1926. Why is there no National Cyber Security Board?

Cyber libertarianism refers to the belief that individuals should be at liberty to pursue their own tastes and interests online. Cyber libertarianism is a common attitude in the tech community; "regulation stifles innovation" is the prevailing mantra. One could imagine a similar attitude being applied to the car industry, but history has shown that regulation and innovation can co-exist. The tech community has not been able to address the cybersecurity situation on its own; it is time to get governments involved, via laws and regulations. Numerous issues will have to be debated and resolved, but we must accept, I believe, that the cybersecurity problem will not be resolved by the market.

Follow me on Facebook, Google+, and Twitter.

Moshe Y. Vardi, EDITOR-IN-CHIEF

Code 2018: Updating the ACM Code of Ethics

The ACM Code of Ethics and Professional Conduct (The Code) outlines fundamental ethical considerations to which all ACM members are expected to adhere.

The Code consists of principles for personal responsibility and guidelines for dealing with many issues computing professionals are likely to face. The Code is intended to serve as a basis for ethical decision making in the conduct of professional work, and includes considerations for individuals in leadership roles.

The ACM Committee on Professional Ethics (COPE) is updating the ACM Code of Ethics and would like your input.

The current ACM Code of Ethics was adopted in 1992, and much has changed in the 25 years since then. We are updating The Code to reflect the shifts in both technology and society.

The 2018 Code is meant to be an update of The Code, not a wholesale revision. We are particularly concerned about possible blind spots or anachronisms that may have resulted from changes in technology or the profession since 1992.

You can help define what it means to be a good computing professional by contributing to the Code 2018 project.

We have completed two drafts of suggested updates to The Code, and we need your input as we begin to work on the final draft.

Get Involved! To review the drafts and to submit your comments, visit: https://code2018.acm.org/discuss

acm *Committee on Professional Ethics*

https://ethics.acm.org

DOI:10.1145/3067094

Vinton G. Cerf

Can Liberty Survive the Digital Age?

As I write this, I am preparing to participate in a Princeton-Fung Global Forum on the topic of the title of this column. It is taking place in cooperation with the Humboldt Institute in the historically significant city of Berlin. The role of digital technology in society has never been more visible than in the unexpected results of the U.S. 2016 Presidential election and the U.K. vote to exit from the European Union ("Brexit"). Online social media have provided a megaphone for voices that might not have been heard except in limited circles. New terms have been introduced into the vocabulary such as "alternative facts" and "fake news." The Internet is not the only path through which these phenomena have propagated, but online social media have demonstrated a triggering capacity beyond earlier expectations. The so-called "Arab Spring" a few years ago also illustrated the collaborative and even coercive power of digital social media, alarming authoritarian regimes, and triggering Internet shutdowns.

It seems timely to explore this question, especially as efforts continue to bring the 50% of the world that is not yet online into parity with the 50% already there. On the positive side, there are many voices that would never be heard were it not for the amplifying power of the Internet; voices crying out for social justice, economic and educational opportunity. That same amplifying effect, however, gives visibility to deliberate (or ignorant) misinformation, hate speech, incitement to violence, and advocacy of terrorism. Naïve Internauts and those unable or unwilling to think critically about what they see and hear, may well accept as valid, bogus and ill-motivated assertions aimed at nefarious objectives and insidious undermining of stable society.

Technical means are of limited value in this arena, although they have proven useful against spam (unsolicited email), scams, malware propagation, and resistance to various forms of digital attack. Social norms, education, and tolerance for diverse views may be critical elements of a response to the challenges that the digital age places on liberty.

To make matters more complex, the Internet and the World Wide Web are transnational phenomena. Information flows do not stop for inspection at national boundaries nor is it clear they should but this makes the challenge of coping with misinformation all the harder. One might hope that our societies would value freedom of expression and tolerant critical thinking that evaluates content and rejects or accepts it based on widely held social norms. The problem with that formulation is that history teaches that social norms can be enormously harmful. One has only to look to history for lessons of slavery, the Holocaust, and Apartheid to realize that reliance on social norms may not produce a fair and equitable society. The so-called "bubble effect" found in social networks only exacerbates the echo chamber phenomenon. Confirmation bias is a well-known problem even in scientific circles where respect for data and its potential to disrupt accepted theory is fundamental to progress.

As I write, the Princeton-Fung Forum is about to get underway, so I do not have solutions or conclusion to offer nor am I confident that solutions will emerge from these discussions. What I am certain of, however, is that it is vital to have these discussions. To wrestle with the problems that widespread access to the mechanisms of information production and consumption appear to pose seems an inescapable responsibility for the creators and users of modern digital technology.

Can liberty truly survive the Digital Age? We won't know the answer unless we try to find ways to assure a positive outcome. We must not only have more and better information to combat bad and misleading information, but we must want to discover that information and to take the time and trouble to assess its merits. In the past, we relied on high-quality journalism with its exercise of responsible editorial management. Today this is becoming increasingly difficult with abundant sources of opinion masquerading as journalism. We must learn how to become our own editors in the same sense that we became our own telephone operators with the advent of direct distance dialing.

The technical community has the opportunity to produce tools that can be used by Internauts everywhere to separate quality information from dross, but the application of those tools falls to individual users willing to exercise critical thinking to get at the facts. Will liberty survive the Digital Age? Yes, I think it can, but only if we make it so. C

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

BLOG@CACM

The *Communications* Web site, http://cacm.acm.org, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

twitter
Follow us on Twitter at http://twitter.com/blogCACM

DOI:10.1145/3057284 http://cacm.acm.org/blogs/blog-cacm

Ending Null Pointer Crashes

Void safety, says Bertrand Meyer, relies on type declarations and static analysis.

Bertrand Meyer

Null-Pointer Crashes, No More

http://bit.ly/2i6w0nz
December 20, 2016

As an earlier article[5] emphasized, code matters; so do programming languages. While Eiffel is best known for its Design by Contract techniques, they are only part of a systematic design all focused on enabling developers to realize the best of their abilities—and eradicate from their code the sources of crashes and buggy behavior.

Talking about sources of crashes, one of the principal plagues of modern programs is null-pointer dereferencing. This term denotes what happens when you call *x.f*, meaning apply *f* (a field access or an operation) to the object that *x* references. If you want to define meaningful data structures, you need to allow "null," also known as Nil and Void, as one of the possible values for reference variables (for example, to terminate linked structures: the "next" field of the last list element must be null, to indicate there is no next element). But then you should make sure that *x.f* never gets called for null *x*, since there is in that case no object to which we can apply *f*.

The problem is particularly acute in object-oriented programming languages, where *x.f* is the major computational mechanism. Every single execution of this construct (how many billions of them occurred in running programs around the world since you started reading this?) faces that risk. Compilers for many languages catch other errors of a similar nature—particularly type errors, such as assigning the wrong kind of value to a variable—but they do nothing about prohibiting null pointer dereferencing.

This fundamental brittleness threatens the execution of most programs running today. Calling it a "billion-dollar mistake" as Tony Hoare did[1] is not an exaggeration. In his recent Ph.D. thesis[2], Alexander Kogtenkov surveyed the null-pointer-derefencing bugs in the Common Vulnerabilities and Exposures (CVE) database, the reference repository of information about Internet attacks. The resulting chart, showing the numbers per year, is edifying:

Beyond the numbers stand real examples, often hair-raising. The description of vulnerability CVE-2016-9113 (http://bit.ly/2mafdkJ) states:

> *There is a NULL pointer dereference in function imagetobmp of convertbmp.c:980 of OpenJPEG 2.1.2. image->comps[0].data is not assigned a value after initialization(NULL). Impact is Denial of Service.*

Yes, that is for the JPEG standard. Try not think of it when you upload your latest pictures. Just for one month (November 2016), the CVE database contains null pointer vulnerabilities affecting products of the Gotha of the IT industry, from Google (http://bit.ly/2mfdAD2) and Microsoft (http://bit.ly/2muJImD) ("*theoretically everyone could crash a server with just a single specifically crafted packet*") to Red Hat (http://red.ht/2lXB5xS) and Cisco (http://bit.ly/2mMcueo). The entry for an NVIDIA example (at http://bit.ly/2lUREf8) explains:

> *For the NVIDIA Quadro, NVS, and GeForce products, NVIDIA Windows GPU Display Driver R340 before 342.00 and R375 before 375.63 contains a vulnerability in the kernel mode layer (nvlddmkm.sys) handler where a NULL pointer dereference caused by invalid user input may lead to denial of service or potential escalation of privileges.*

We keep hearing complaints that "the Internet was not designed with security in mind." What if the problem had far less to do with the design (TCP/IP is brilliant) than with the languages that people use to write tools implementing these protocols?

In Eiffel, we decided that the situation was no longer tolerable. After the language had eradicated unsafe casts through the type system, memory management errors through garbage collection and data races through the SCOOP concurrency mechanism, null pointer dereferencing was the remaining dragon to slay. Today Eiffel is *void-safe*: a null pointer dereference can simply not happen. By accepting your program, the compiler guarantees that every single execution of every single *x.f* will find *x* attached to an actual object, rather than void.

How do we do this? I am not going to describe the void-safe mechanism in detail here, referring instead to the online documentation[6], with the warning it is still being improved. But I can give the basic ideas. The original article describing void safety (and giving credit to other languages for some of the original ideas) was a keynote at ECOOP in 2005[3]. Revisiting the solution some years later, I wrote[4]:

> *Devising, refining, and documenting the concept behind the mechanism presented here took a few weeks. The engineering took four years.*

That was optimistic. Seven more years later, the "engineering" continues. It is not a matter of ensuring void safety; the mechanism was essentially sound from the beginning. The continued fine-tuning has to do with facilitating the programmer's task. Any mechanism that avoids bugs—another example is static typing—buys safety and reliability at a possible cost in expressiveness: you have to prohibit harmful schemes (otherwise you would not avoid any bugs), but you do not want to prohibit useful schemes or make them too awkward to express (otherwise it is very easy to remove bugs: just reject all programs!) or make them too awkward to express. The "engineering" consists of ever more sophisticated static analysis, through which the compiler can accept safe cases that simplistic rules would reject.

In practice, the difficulty of fine-tunign void safety mostly involve the *initialization* of objects. While the details of void safety can be elaborate, the essential idea is simple: the mechanism relies on *type declarations* and *static analysis*.

The void-safe type system introduces a distinction between "attached" and "detachable" types. If you declare a variable *p1* as just of type (for example) *PERSON* it can never be void: its value will always be a reference to an object of that type; *p1* is "attached." This is the default. If you want *p2* to accept a void value you will declare it as **detachable** *PERSON*. Simple compile-time consistency rules support this distinction: you can assign *p1* to *p2*, but not the other way around. They ensure an "attached" declaration is truthful: at runtime, *p1* will always be non-void. That is a formal guarantee from the compiler.

The static analysis produces more such guarantees, without particular actions from the programmers as long as the code is safe. For example, if you write

```
if p2 /= Void then p2.f end
```

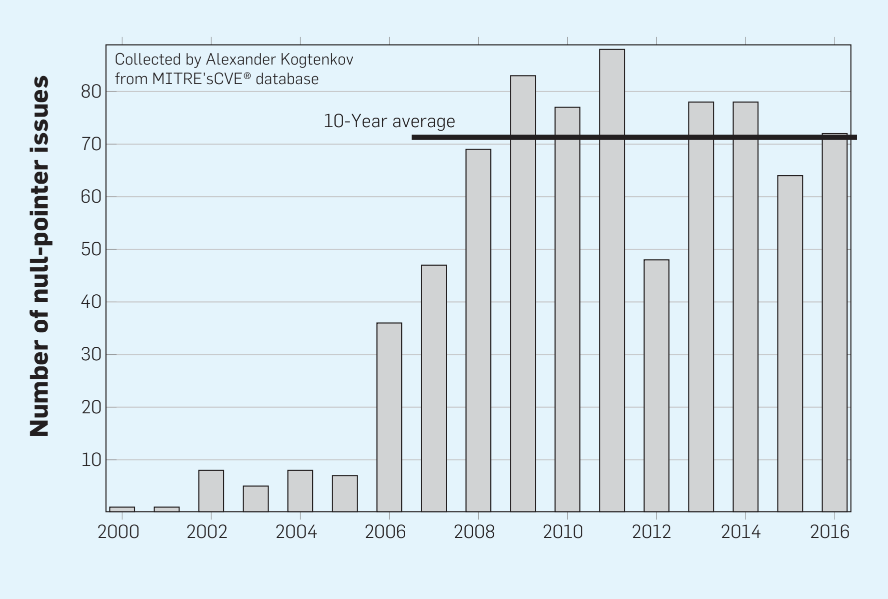
we know that things are OK. (Well, under certain conditions. In concurrent programming, for example, we must be sure that no other thread running in parallel can make *p2* void between the time we test it and the time we apply *f*. The rules take care of these conditions.)

The actual definition cannot, of course, say that "the compiler" will recognize safe cases and reject unsafe ones. We cannot just entrust the safety of our program to the inner workings of a tool (even open-source tools like the existing Eiffel compilers). Besides, there is more than just one compiler. Instead, the definition of void safety uses a set of clear and precise rules, known as Certified Attachment Patterns (CAPs), which compilers must implement. The preceding example is just one such CAP. A formal model backed by mechanized proofs (using the Isabelle/HOL proof tool) provides[2] solid evidence of the soundness of these rules, including the delicate parts about initialization.

Void safety has been here for several years, and no one who has used it wants to go back. (The conversion to voided safety of older, non-void-safe projects is not as painless.) Writing void-safe code quickly becomes second nature.

And what about your code: are you certain it can never produce a null-pointer dereference?

Null pointer issues (such as null pointer dereferencing) in Common Vulnerabilities and Exposures Database.



References

1. Hoare, C.A.R., Null References: The Billion-Dollar Mistake, August 25, 2009, http://bit.ly/2lAhgeP
2. Kogtenkov, A., Void Safety, ETH Zurich Ph.D. thesis, January 2017, http://se.inf.ethz.ch/people/kogtenkov/thesis.pdf.
3. Meyer, B., Attached Types and their Application to Three Open Problems of Object-Oriented Programming, in ECOOP 2005 (*Proceedings of European Conference on Object-Oriented Programming*, Edinburgh, 25-29 July 2005), ed. Andrew Black, Lecture Notes in Computer Science 3586, Springer, 2005, pages 1-32, http://bit.ly/2muJ8Ff
4. Meyer, B., Kogtenkov, A., and Stapf, E.: *Avoid a Void: The Eradication of Null Dereferencing*, in *Reflections on the Work of C.A.R. Hoare*, eds. C. B. Jones, A.W. Roscoe and K.R. Wood, Springer, 2010, pages 189-211, http://bit.ly/2lsNfNO
5. Meyer, B., Those Who Say Code Does Not Matter, *CACM*, April 15, 2014, http://bit.ly/1mNqout
6. Void safety documentation at eiffel.org: http://bit.ly/2lsS2xZ

Bertrand Meyer is a professor of software engineering at Politecno di Milano and Innopolis University.

N news

Science | DOI:10.1145/3057735

Esther Shein

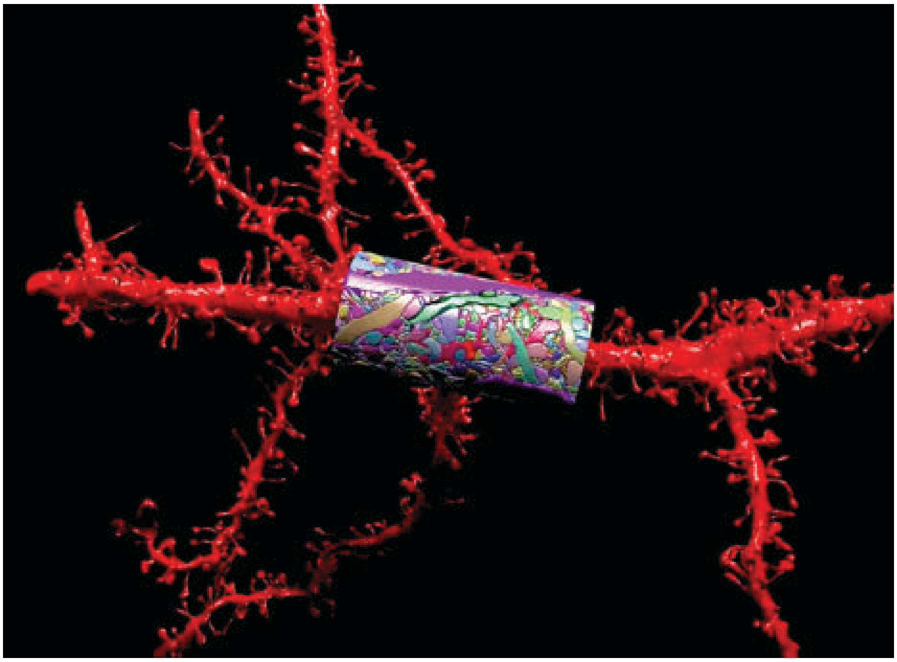
Combating Cancer With Data

Supercomputers will sift massive amounts of data in search of therapies that work.

FOR DECADES, SCIENTISTS have worked toward the 'holy grail' of finding a cure for cancer. While significant progress has been made, their efforts have often been worked on as individual entities. Now, as organizations of all kinds seek to put the massive amounts of data they take in to good use, so, too, are the health care industry and the U.S. federal government.

The National Cancer Institute (NCI) and the U.S. Department of Energy (DOE) are collaborating on three pilot projects that involve using more intense high-performance computing at the exascale level, which is the push toward making a billion billion calculations per second (or 50 times faster than today's supercomputers), also known as exaFLOPS (a quintillion, 10^{18}, floating point operations per second). The goal is to take years of data and crunch it to come up with better, more effective cancer treatments.

The DOE had been working on building computing infrastructure capable of handling big data and entered into discussions with the NCI, which houses massive amounts of patient data. The two organizations realized there were synergies between their efforts and that they should collaborate.



Researchers used scanning electron microscope images of nanometers-thick mouse brain slices to reconstruct cells into a neocortex structure (center), whose various cell types appear in different colors.

IMAGE COURTESY OF ARGONNE NATIONAL LABORATORY

The time is right for this particular collaboration because of the application of advanced technologies like next-generation sequencing, says Warren Kibbe, director of the NCI Center for Biomedical Informatics and Information Technology. In addition, data is becoming more readily available from vast repositories, and analytics and machine learning tools are making it possible to analyze the data and make better sense of it.

Says Kibbe, "There is ever-better instrumentation and data acquisition from that instrumentation, such as using cryoEM (cryo-electron microscopy) to generate structural data in biology, that lets us now look at molecules that up until now have been very difficult to look at.” Recently, he adds, “there’s been a tremendous infusion of technology in biology,” enabling, for example, the ability to interrogate a tissue and determine the types of cells in the tissue and their spatial organization.

Many big challenges still exist, such as learning how individual cells work together in the tumor micro-environment and how they contribute to the overall aggressiveness of cancer and its ability to resist therapies, Kibbe adds.

The opportunity to work with the DOE meant exposure to a tremendous amount of computational expertise and thinking about problems in deep learning and natural language processing (NLP), as well as being able to do very detailed simulations, he says. Taking the available cancer data and using it to build mechanistically informed models and predictive models will enable researchers to better understand, as they perturb a particular cell, how that perturbation is going to impact the tissue and the biological system. It will also tell researchers whether they can “do a better job providing patients with optimal therapies based on the modeling.”

For the NCI/DOE collaboration, the goal is not understanding individual cells and tissues, but whether researchers can glean from a huge population how patients respond when they are given a particular therapy. “That’s a data aggregation problem and a natural language processing problem,’’ Kibbe says. “The DOE has a lot of expertise in looking not only at energy grids, but thinking about integrating data from a number of different sources and technologies, and building up simulations and models.”

One pilot by Argonne National Laboratory focuses on deep learning and building predictive models for drug treatment response using different cell lines and patient-derived xenografts (tissue grafts from a donor of a different species than the recipient). “We’re trying to build models where we can predict where tumors we haven’t screened will respond to a drug,” explains Rick Stevens, associate laboratory director for computing, environment, and life sciences research at Argonne, who is spearheading the deep learning pilot. This is the underlying concept of precision medicine.

“A wonderful feature of the artificial intelligence community is that it’s very open. You have collaborations that span companies that are competing with each other.”

Tumor cells have thousands of different types of molecules and tens of thousands of genes that change all the time, so there are fundamental points that researchers don’t understand, Stevens explains. Building a model based on principles of what is happening in cancer cells is incomplete; if a researcher tried to make predictions of how a cancer cell will respond without taking into consideration the properties of the treatments, it wouldn’t be as effective. That’s where the team hopes deep learning applied to drug combination therapies will be useful.

A second pilot, at Lawrence Livermore National Laboratory, is aimed at understanding the predictive paths in the Ras cancer gene, mutations of which are responsible for about 30% of all cancers, Stevens says. Work there is also focused on the oncogene which, when mutated, becomes the driver for causing cancer. “It’s one of the core targets we’re trying to understand [as well as] how to drug it,’’ says Stevens. “It’s stuck in the ‘on’ position; it’s like a switch and it tells your cells when to divide.”

A third pilot, under way at Oak Ridge National Laboratory, is mining data from millions of patient records in search of large-scale patterns to optimize drug treatments. The pilot is working with the Surveillance, Epidemiology and End Results (SEER) Registries, which NCI has used since 1974 to assess the incidence and outcomes for cancer patients across the country and covers roughly 30% of the U.S. population, says Stevens. However, the challenge is that because it was built over 40 years ago, it “has seen a lot of technologies, and the hope is we can transform the SEER Registries into something that has very different characteristics” using NLP and deep learning features.

This is where the partnership with DOE will be especially valuable, says Kibbe, because the department has a lot of expertise working with sensor networks and data aggregation interrogation and analysis.

The common thread among all three pilots is that each has a deep learning component to them, Stevens says. To fund the initiatives, he and his co-investigators received $5 million in fall 2016 from the Exascale Computing Project (ECP) to build a deep neural network code called the CANcer Distributed Learning Environment (CANDLE). This year, Argonne, Lawrence Livermore, and Oak Ridge all will deploy their highest-performing supercomputers available and the teams will use these systems to start evaluating existing open source software from various vendors and test machine learning capabilities. That way, Stevens notes, they won’t have to reinvent the wheel.

“We’ll add what we need on top of the frameworks and make it possible to use the large-scale hardware we have and feed it back into the open source community,” Stevens says. “A wonderful feature of the artificial intelligence community is that it’s very open. You have collaborations that span companies that are competing with each other,” including Microsoft, Google, and Facebook.

The teams working on the three pilots will “run big benchmark problems on the DoE hardware,” and will have the first code release that can serve all three pilots and eventually other application areas in the summer, he says.

One of the problems, in Stevens’ case, is a classification problem, in which tumor expression data, known as SNP (single nucleotide polymorphisms) data, is used to try to determine what type of cancer is being studied from the SNPs alone. “That hasn’t been done before; it’s related, but not the same to classification of gene expression,’’ he says. And there are several other problems as well, including trying to predict the response to an individual drug based on its formula and profile, and the auto encoder problem, in which a network is trained to learn the compressed representation of a drug structure, for example, and then has to be trained to accurately reproduce the input so the team can build an improved algorithm.

The benchmarks will change over time, but they are a way to develop a common language among the vendors and the teams working on the pilots, Stevens says.

Once the first iteration of the model has been built and validated, it should be able to analyze tumor information from a newly diagnosed cancer patient and predict which drug will be the most effective at attacking the tumor.

Meanwhile, to help foster existing collaborations and pursue new ones, the first of a series of meetings was held in July 2016. The Frontiers of Predictive Oncology and Computing meeting focused on predictive oncology and computing in a few areas of interest in NCI/DOE collaboration: basic biology, pre-clinical, clinical applications and computing, says Eric Stahlberg, a contractor working on the high-performance computing strategy within the Data Science and Information Technology Program at the Frederick National Laboratory for Cancer Research in Rockville, MD.

“Efforts at the frontier of pre-clinical predictive oncology … included developing new models using patient-derived xenografts and predicting drug efficacy through regulatory networks,’’ Stahlberg says. Other areas of focus were how to gain better insights into Ras-related cancers, gathering quality data for use in predictive model development, and improving the SEER database.

“The meeting attendees were very enthusiastic about the prospects for improving cancer patient outcomes with increased use of computing,’’ Stahlberg says. That said, “One of the largest challenges exists in developing interoperability among solutions used in predictive oncology.” Others include gathering consistent data and having enough data to understand the complexity of individual cells, he says.

Since the conference, further progress has been made in yet another collaboration: the public-private partnership for Accelerating Therapeutic Opportunities in Medicine (ATOM) involving GlaxoSmithKline, the DOE, and the NCI, he says. Additionally, “most significantly, the 21st Century Cures Act was just signed into law, setting the stage for a very promising future at the intersection of predictive oncology and computing.”

Several universities also are actively researching ways to tackle big data, which is a big challenge given the tremendous amount of information collected in the life sciences, notes Sunita Chandrasekaran, an assistant professor in the Center for Bioinformatics and Computational Biology at the University of Delaware, and one of the meeting’s organizers.

“Efforts are under way in universities that collaborate with medical research institutes or facilities in order to accelerate such large-scale computations like sequence alignment using accelerators like GPUs (graphics-processing units),’’ she says. “Efforts are also under way to build suitable and portable software algorithms that can adapt to varying input and generate results dynamically adapting to evolving hardware.”

Stevens says what makes it possible now to use data more effectively than several years ago is that researchers have found ways to accelerate deep learning through things like GPUs. “This, coupled with breakthroughs in some methods like convolutional neural networks, has suddenly made deep learning effective on many problems where we have large amounts of training data.”

When the single model has been put into effect, researchers will be able to add more information about cancer cells as well as more information about drugs, “and we would have many more instances of ‘this drug worked this well on a given tumor,’ so many more training pairs between cancers and drugs,” says Stevens.

While acknowledging he hates to make predictions, Kibbe feels confident that “in the next 10 years we should see that many of what are very hard-to-treat cancers will be treated,” and that regardless of where someone lives and what their socioeconomic status is, they will have access to the same level of care.

“I think that’s what will come out of these collaborations and use of computing; as sensors and instrumentation get cheaper and cheaper to implement and become more and more ubiquitous, the hope is there will be a leveling effect on cancer treatment across the country, and perhaps the whole world.”

Perhaps working in collaboration, combined with deep learning and highly advanced computing, will prove to be that holy grail. Kibbe calls the DOE/NCI partnership unique in that two very different cultures are working together as a team. While everyone is excited about their individual projects, he says, they are also excited about their joint mission of creating a workforce that has both biomedical knowledge and computational expertise.

“That side of the collaboration is going to continue to pay dividends for as long as we have computation in biomedical research, which I hope is forever.”

Further Reading

Davis, J.
**Can Big Data Help Cure Cancer? *InformationWeek*, July 17, 2016.
http://www.informationweek.com/big-data/big-data-analytics/can-big-data-help-cure-cancer-/d/d-id/1326295**

Agus, D.B.
**Giving Up Your Data to Cure Disease, *The New York Times*, Feb. 6, 2016.
https://www.nytimes.com/2016/02/07/opinion/sunday/give-up-your-data-to-cure-disease.html?_r=0**

Panda, B.
**Big Data and Cancer Research. Springer, Oct. 13, 2016.
http://link.springer.com/chapter/10.1007%2F978-81-322-3628-3_14**

Cho, W.C.
**Big Data for Cancer Research, *Clinical Medicine Insights: Oncology.* v.9; 2015 PMC4697768.
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4697768/**

Reed, D.A., and Dongarra, J.
**Exascale Computing and Big Data, *Communications of the ACM*, Volume 58, Issue 7, July 2015, pp. 56-68.
http://dl.acm.org/citation.cfm?id=2699414**

Esther Shein is a freelance technology and business writer based in the Boston area.

Technology | DOI:10.1145/3057740

Samuel Greengard

Making Chips Smarter

Advances in artificial intelligence and machine learning are motivating researchers to design and build new chips to support different computing models.

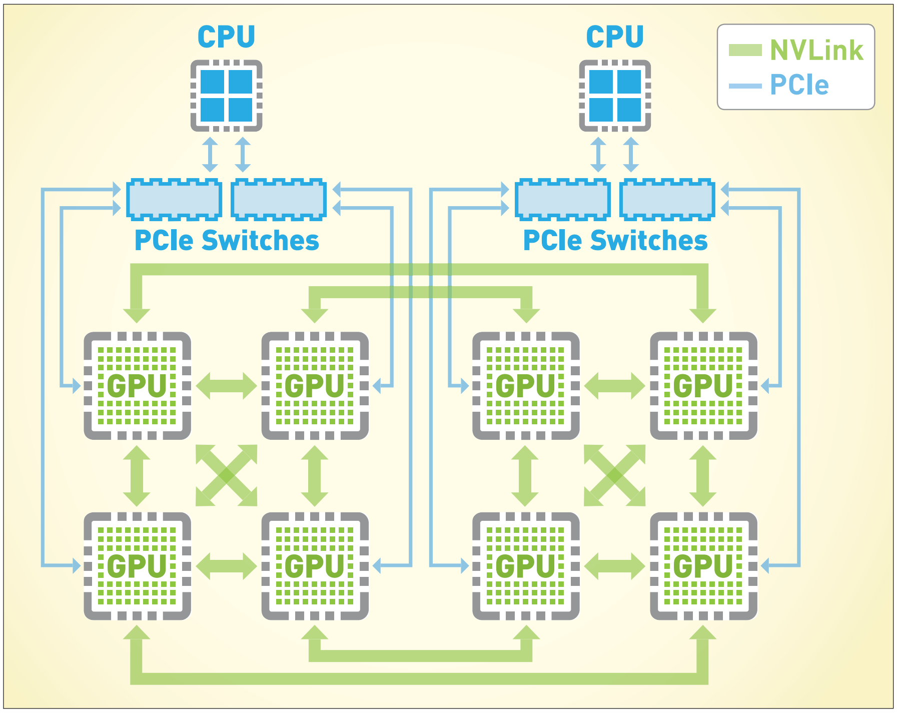
It is no secret that artificial intelligence (AI) and machine learning have advanced radically over the last decade, yet somewhere between better algorithms and faster processors lies the increasingly important task of engineering systems for maximum performance—and producing better results.

The problem for now, says Nidhi Chappell, director of machine learning in the Datacenter Group at Intel, is that "AI experts spend far too much time preprocessing code and data, iterating on models and parameters, waiting for training to converge, and experimenting with deployment models. Each step along the way is either too labor- and/or compute-intensive."

The research and development community—spearheaded by companies such as Nvidia, Microsoft, Baidu, Google, Facebook, Amazon, and Intel—is now taking direct aim at the challenge. Teams are experimenting, developing, and even implementing new chip designs, interconnects, and systems to boldly go where AI, deep learning, and machine learning have not gone before. Over the next few years, these developments could have a major impact—even a revolutionary effect—on an array of fields: automated driving, drug discovery, personalized medicine, intelligent assistants, robotics, big data analytics, computer security, and much more. They could deliver faster and better processing for important tasks related to speech, vision, and contextual searching.

Specialized chips can significantly increase performance for fixed-function workloads, because they include everything needed specifically for the task at hand and nothing more. Yet, the task is not without its challenges.

For one thing, there's no clear idea about how to use silicon to accelerate AI. Most chip designs and systems are still in the early stages of research, development, or deployment.



The design of the NVIDIA NVLink Hybrid Cube Mesh, which connects eight graphics processing units, each with 15 billion transistors.

IMAGE BY ANDRIJ BORYS ASSOCIATES, BASED ON SCHEMATIC FROM NVIDIA BLOG

For another, there's no single design, approach, or method that works well for every situation or AI-based framework.

One thing that is perfectly clear: AI and machine learning frameworks are advancing rapidly. Says Eric Chung, a researcher at Microsoft Research: "We're seeing an escalating, insatiable demand for this kind of technology."

Beyond the GPU

The quest for faster and better processing in AI is nothing new. In recent years, graphical processing units (GPUs) have become the technology of choice for supporting the neural networks that support AI, deep learning, and machine learning. The reason is simple, even if the underlying technology is complex: GPUs, which were originally invented to improve graphics processing on computers, execute specific tasks faster than conventional central processing units (CPUs). Yet, a specialized design is not ideal for every application or situation. For instance, a search engine such as Bing or Google has very different requirements than the speech processing used on a smartphone, or the visual processing that takes place in an automated vehicle or in the cloud. To varying degrees, systems must support both training and delivering real-time information and controls.

In the quest to boost performance in these systems, designers and engineers are leaving no idea unexamined. However, all the research revolves around a key goal: "Specialized AI chips will deliver better performance than either CPUs or GPUs. This will undoubtedly shift the AI compute [framework] moving forward," Chappell explains. In the real world, these boutique chips would greatly reduce training requirements in neural networks, in some cases from days or weeks to hours or minutes. This has the potential to not only improve performance but also slash costs for companies developing AI, deep learning, and machine learning systems. The result would be faster and better visual recognition in automated vehicles, or the ability to reprocess millions of scans for potentially missed markers in healthcare or pharma.

The focus on boutique chips and better AI computation is leading researchers down several avenues. These include improvements in GPUs as well as work on other technologies such as field programmable gate arrays (FPGAs), Tensor Processing Units (TPUs), and other chip systems and architectures that match specific AI and machine learning requirements. These initiatives, says Bryan Catanzaro, vice president of Applied Deep Learning Research at Nvidia, point in the same general direction: "The objective is to build computation platforms that deliver the performance and energy efficiency needed to build AI with a level of accuracy that isn't possible today."

GPUs, for instance, already deliver superior processor-to-memory bandwidth and they can be applied to many tasks and workloads in the AI arena, including visual and speech processing. The appeal of GPUs revolves around providing greater floating-point operations per second (FLOPs) using fewer watts of electricity, and the ability to extend the energy advantage by supporting 16-bit floating point numbers, which are more power- and energy-efficient than single-precision (32-bit) or double-precision (64-bit) floating point numbers. What is more, GPUs are quite scalable. The Nvidia Tesla P100 chip, which packs 15 billion transistors into a silicon chip, delivers extremely high throughput on AI workloads associated with deep learning.

However, as Moore's Law reaches physical barriers, the technology must evolve further. For now, "There are a lot of ways to customize processor architectures for deep learning," Catanzaro says. Among these: improving efficiency on deep learning specific workloads, and better integration between throughput-oriented GPU and latency-oriented CPU. For instance, Nvidia has introduced a specialized server called DGX-1, which uses eight Tesla P100 processors to deliver 170 teraflops of compute for neural network training. The system also uses a fast interconnect between GPUs called NVLink, which the company claims allows up to 12 times faster data sharing than traditional PCIe interconnects.

> "The objective is to build computation platforms that deliver the performance and energy efficiency needed to build AI with a level of accuracy that isn't possible today."

"There is still an opportunity for considerable innovation in this space," he says.

New Models Emerge

Other approaches are also ushering in significant gains. For example, Google's Tensor Processing Unit (TPU) is a custom application-specific integrated circuit (ASIC) that is specifically designed for AI applications such as speech processing and street-view mapping and navigation. It has been used in Google's datacenters for more than 18 months. A big benefit is that the chip is optimized for reduced computational precision. This translates into fewer transistors per operation and the ability to squeeze more operations per second into the chip, which results in better-optimized performance per watt and an ability to use more sophisticated and powerful machine learning models—while applying the results more quickly.

Another technology aimed at advancing AI and machine learning is Microsoft's Project Catapult, which uses field programmable gate arrays (FPGAs) that underpin the widely used Bing search engine, as well as the Azure cloud. This allows teams to implement algorithms directly onto hardware, rather than potentially less-efficient software. Chung says the FPGAs' performance exceeds that of CPUs while retaining flexibility and allowing production systems to operate at hyperscale. He describes the technology as "programmable silicon."

To be sure, energy-efficient FPGAs satisfy an important requirement when deploying accelerators at hyperscale in power-constrained datacenters. "The system delivers a scalable, uniform pool of resources independent from CPUs. For instance, our cloud allows us to allocate few or many FPGAs as a single hardware service," he explains. This, ultimately, allows Microsoft to "scale up models seamlessly to a large number of nodes. The result is extremely high throughput with very low latency."

FPGAs are, in fact, highly flexible chips that achieve higher performance and better energy efficiency with reduced numerical precision. "Each computational operation gets more efficient on the FPGA with the fewer bits you use," Chung explains. The current generation of these Intel chips, known as Stratix V FPGAs, will evolve into more advanced versions, including Arria 10 and Stratix 10, he notes. They will introduce additional speed and efficiencies.

"With the technology, we can build custom pipelines that are tailored to specific algorithms and models." Chung says. In fact, Microsoft has reached a point where developers can deploy models rapidly, and without underlying technical expertise about the machine learning framework. "The appeal is the high level of flexibility. It can be reprogrammed for different AI models and tasks," Chung notes. In fact, the FPGAs can be reprogrammed on the fly to respond to advances in artificial intelligence or different datacenter requirements. A process that previously could take two years or more, now can take place in minutes.

Finally, Intel is introducing Nervana, a technology that aims to "deliver unprecedented compute density and high bandwidth interconnect for seamless model parallelism," Chappell says. The technology will focus primarily on multipliers and local memory, and skip elements such as caches that are required for graphics processing but not for deep learning. It also features isolated pipelines for computation and data management, as well as High Bandwidth Memory (HBM) to accelerate data movement. Nervana, which Intel expects to introduce during the first half of this year, will "deliver sustained performance near theoretical maximum throughput," he adds. It also includes 12 bidirectional high-bandwidth links, enabling multiple interconnected engines for seamless scalability, a key requirement for increased performance through scale.

Into the Future

An intriguing aspect of emerging chip designs for AI, deep learning, and machine learning is the fact that low-precision chip designs increasingly prevail. In many cases, reduced-precision processors conform better to neuromorphic compute platforms and accelerate the deployment and possibly training of deep learning algorithms. Simply put: they can produce similar results while consuming less power, in some cases by a factor of 100. While algorithms running on today's digital processors require high numerical precision, the same algorithms operating on low precision chips in a neural net excel, because these systems adapt dynamically by examining data in a more relational and contextual way (and are less sensitive to rounding errors).

This makes the technology perfect for an array of machine learning tasks and technologies, including drones; automated vehicles; intelligent personal assistants such as Amazon's Alexa, Microsoft's Cortana, or Apple's Siri; photo and image recognition systems, and search engines, including general services like Bing and Google but also those used by retailers, online travel agencies, and others. It also supports advanced functionality like real-time speech-to-text transcription and language translations.

Microsoft has reached a point where developers can deploy models rapidly, without underlying technical expertise about the machine learning framework.

In the end, says Gregory Diamos, a senior researcher at Baidu, specialized machine learning chips have the potential to change the stakes and usher in an era of even greater breakthroughs. "Machine learning has already made tremendous progress," he says. "Specialized chips and systems will continue to close the gap between computers and human performance."

Further Reading

Caulfield, A., Chung, E., Putnam, A., Angepat, H., Fowers, J., Haselman, M., Heil, S., Humphrey, M., Kaur, P., Kim, J.Y., Lo, D., Massengill, T., Ovtcharov, K., Papamichael, M., Woods, L., Lanka, S., Chiou, D., and Burger, D.
A Cloud-Scale Acceleration Architecture, October 15, 2016. *Proceedings of the 49th Annual IEEE/ACM International Symposium on Microarchitecture*, IEEE Computer Society. https://www.microsoft.com/en-us/research/publication/configurable-cloud-acceleration/

Samel, B., Mahajan, S., and Ingole, A.M.
GPU Computing and Its Applications, *International Research Journal of Engineering and Technology (IRJET)*. Volume: 03 Issue: 04, Apr-2016. https://www.irjet.net/archives/V3/i4/IRJET-V3I4357.pdf

Shafiee, A., Nag, A., Muralimanohar, N., Balasubramonian, R., Strachan, J.P., Hu, M., Williams, S.R., and Srikumar, V.
ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars, *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pp. 14-26, 2016, ISSN 1063-6897. http://ieeexplore.ieee.org/document/7446049/citations

Shirahata, K., Tomita, Y., and Ike, A.
Memory reduction method for deep neural network training, *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2016, pp. 1-6. doi: 10.1109/MLSP.2016.7738869. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7738869&isnumber=7738802

Samuel Greengard is an author and journalist based in West Linn, OR.

ACM Member News

FINDING THE INTERSECTION OF MATH AND LANGUAGE

When Bulgarian-born Dragomir Radev, professor of computer science at the University of Michigan, studied computer science as an undergraduate at the Technical University of Sofia, "I was interested in math and languages; French, Russian, and English. I was not sure how to combine those two interests," Radev explains. "When the first personal computers came around, I thought it would be a good way to combine my interests."

Radev completed his undergraduate degree at the University of Maine at Orono, before going on to earn a Ph.D. in computer science at Columbia University in New York in 1999 (while serving as an adjunct assistant professor in the department of computer science). His focus was on natural language processing and computational linguistics, working on algorithms to teach human languages to computers.

Even before graduating, Radev was hired by IBM in 1998 to work on the team that built the first question/answer system at the company's Thomas J. Watson Research Center in Hawthorne, NY. "After a year-and-a-half at IBM, I started at the University of Michigan" in January 2000, he adds, "and I have been there since."

Radev now is involved with building spoken dialog systems for student advising, and he serves on the executive committee at the Association for Computational Linguistics, an organization for those working on problems involving natural language and computation. He has also served as co-chair of the North American Computational Linguistics Olympiad (NACLO), in which thousands of high school students in the U,S, and Canada compete to solve problems in natural language processing and computational linguistics.

—John Delaney

Society | DOI:10.1145/3057738

Keith Kirkpatrick

Bionics in Competition

Developers of innovative assistive devices compete as a means of networking with each other.

MOST PHYSICAL COMPETITIONS are based around the idea of participants pushing themselves physically, demonstrating to the world that they are the fastest, strongest, or otherwise physically gifted. For those with significant physical disabilities or injuries, however, simply accomplishing basic everyday tasks can be an Olympic-level feat.

That's where Cybathlon, a new competition designed to promote innovative assistive devices, may accomplish two goals: providing a competitive forum for disabled athletes, and highlighting the specific advances that are being made in robotic assistive aids designed to help those with significant physical disabilities.

Conceived and developed by Switzerland's ETH Zurich (a science and research university) and National Centre of Competence in Research (NCCR) Robotics professor Robert Riener, the first iteration of Cybathlon took place last October in Zurich. During this international competition, 66 technical teams (comprised of one pilot or operator, along with a number of researchers and scientists) from 25 countries came together to compete in six different disciplines of events.

Each team consisted of at least one technology provider, which was a member of a research lab or a company, and at least one pilot, a person with a specified level of disability that is being managed by using technology developed by the team. The overall competition consisted of six so-called "disciplines," each consisting of tasks that must be completed in the fastest time possible, and ahead of all other teams.

Silke Pan of Team PolyWalk EPFL in the powered exoskeleton race.

While each team can compete in any of the disciplines, which include the brain-computer interface race; the functional electrical stimulation bicycle race; the powered-arm prosthesis race; the powered-leg prosthesis race; the powered-exoskeleton race, and the powered-wheelchair race, only one pilot can participate per team per discipline.

The disciplines are designed to showcase the technology that can be used to improve the lives of those living with a disability, by creating specific challenges that mimic the obstacles that are faced by such people every day. "Compared to the Paralympics, who are searching for the strongest and fastest, we are searching for those pilots who are most skilled to use a device for daily life activities," Riener explains. "We do not consider our event as a sport, though the participants have to train, and they have to perform well."

For example, the powered-wheelchair race includes six hurdles, such as entering a building with thresholds or narrow doorways, or crossing uneven pavement, that must be completed in as little time as possible.

Meanwhile, the brain-computer interface race provides a competition for teams who have developed methods for using brain waves to control avatars in a computer game, which is analogous to brain waves ultimately being used to control objects in the real world. This type of control ultimately will be useful to those with partial or total paralysis.

The powered-arm prosthesis race pits competitors against one another, making them complete daily tasks such as slicing bread or placing silverware on a table using only their prosthetic arms, as quickly and accurately as possible. These fine-motor coordination tasks are technically challenging for those with limb loss or damage, and many of the solutions highlighted in the competition could be further developed in the future for use in the real world.

Unlike other sporting or robotics competitions, the end goal goes beyond establishing a "winner." According to Riener, the goals of Cybathlon are to facilitate conversation between academia and industry, to engender discussion between technology developers and people with disabilities, and to promote the use of robotic assistive aids to the general public.

"A platform like Cybathlon allows people to see what is the state of the art and what is upcoming," says David Langlois, a team leader with Iceland-based prosthetics and orthotics developer Össur, which brought its Rheo Knee, an advanced learning prosthetic device that automatically adapts to the user and the environment, to Cybathlon, and took home the top prize in the powered leg prosthesis race. "This type of event is like going to a car show to see what is new. The only difference is that the manufacturers have to complete a series of mundane tasks to show what their devices are really about."

Furthermore, researchers praised Cybathlon as a platform to showcase interim advances in their work. "Cy-

IMAGE COURTESY OF ETH ZÜRICH/NICOLA PITARO

bathlon sets a deadline and pushes for delivery of the innovation which has happened before," says Knut Lechler, Össur's other team leader. "The Cybathlon provided a platform to show what we have in the pipeline."

Lechler's colleague David Langlois noted that Cybathlon represents a new way for commercial providers such as Össur to market the real-world user benefits of their technologies prior to being released or sold to the public, rather than simply highlighting clinical results or technical specifications. Says Langlois: "You can see Cybathlon as a reversal of the usual innovation competition framework, challenging the manufacturers and innovators to showcase their contribution to the users."

The structure of Cybathlon is also unique in that both the pilots who control the devices and the technology itself are of equal importance.

"Pilots have to show that they can complete a task that is integral to daily life," Riener says. "The device needs the pilot, because it needs someone to control the device."

According to Riener, most of the teams are from universities and other non-profit development groups, though about 25% were for-profit, commercial, or industrial groups. However, Riener says that the types of solutions presented by the corporate teams were generally simpler, but more robust in nature.

"The companies want to develop technology that can be commercialized quickly, and that's why they develop solutions that can be considered to be more practical," Riener says.

On the other hand, many competitors at Cybathlon are academic researchers, such as NeuroCONCISE, a non-profit group that has developed wearable neurotechnology. NeuroCONCISE's solution noninvasively measures and translates brain waves into control signals that permit people to communicate and interact with computers without moving. The group took third place in the brain-computer interface race, and team leader Damien Coyle noted the competitive angle helped motivate and reinforce the team's belief that its work is on the right track.

"This competition was going to really test and raise the bar to see what the technology could achieve," Coyle says. "It also raised awareness among the public about the technologies that are out there, and put us all under pressure" to make sure the work they are doing is viable.

Another team that competed at Cybathlon came from the Florida-based Institute for Human and Machine Cognition, or IHMC, which has been working on an exoskeleton using torque-controlled actuators and powered joints to help people who have been paralyzed, or who have lost a limb. The group competed and earned a silver medal in the powered exoskeleton race, in which the pilot needed to complete six tasks that are common to everyday living.

Team leader Peter Neuhaus said care was taken by the organizers to make sure the tasks would be challenging, yet not so difficult to complete as to be unreasonable. Significant attention was paid to ensuring the tasks were as closely related to real-world scenarios as possible, which meant all technology designs needed to be practically focused, rather than focused on abstract concepts or movements. Indeed, the tasks in the exoskeleton race—getting up from a sofa, walking around obstacles, walking up a ramp to open and walk through a door, walking over stepping stones, walking over an uneven floor, and walking up and down stairs—are tasks likely to be encountered on a regular basis by people with impaired mobility.

"Our research group has been in other types of competitions before," Neuhaus says. "The challenge with competitions is to ensure that the solution to the task advances the research field. The solutions developed for Cybathlon use advances that carry on beyond the competition, and can operate in the real world."

All told, the significant amount of attention paid to Cybathlon—more than 4,600 spectators attended in person, and international media coverage of the event was strong—helped raise awareness of the research being done in universities and among for-profit companies. Participating in Cybathlon is "something you can tell people about," Coyle says. "It's quite a unique thing, and it opened up further avenues for where the technology could go."

Cybathlon is also helpful in eliminating some of the silos that often occur in research and commercial development labs.

"A competition like Cybathlon provides a great insight on what is the current thinking about real-life challenges associated with disabilities," Langlois says. "Furthermore, since there is always a lot of ways to solve these problems and there is no book telling you how to resolve it, a friendly competition between innovators and engineers is always a good way to stimulate creative minds and drive out technology."

According to participants, there aren't any similar events being produced, either in the U.S. or around the world, that aren't affiliated with Riener's group; Cybathlon's close ties with researchers and corporate entities involved in bionic prosthetics and brain research likely has consolidated support around the Cybathlon brand and event. Riener says smaller regional events that license the Cybathlon name may be launched around the world over the next four years, and another major event is slated to take place in Zurich again in 2020.

Interest remains high, as current assistive technology is not yet satisfactory, according to Riener. "The wheelchairs are still too bulky, and can't go over uneven terrain," he says. "The commercially available prosthetic devices are still not powered, which makes it very challenging to climb stairs or walk up ramps." C

Further Reading

Cybathlon Championship for Athletes with Disabilities: http://www.cybathlon.ethz.ch/

Cybathlon 2016 Highlights: https://www.youtube.com/watch?v=KAVcVfKoYwc

Reiner, R.
Cybathlon: A bionics competition for people with disabilities, http://robohub.org/cybathlon-a-bionics-competition-for-people-with-disabilities/

Keith Kirkpatrick is principal of 4K Research & Consulting, LLC, based in Lynbrook, NY.

DOI:10.1145/3061359

TURING AWARD SINCE 1966
CELEBRATING 50 YEARS OF COMPUTING'S GREATEST ACHIEVEMENTS

The Internet of Things

SINCE ITS INAUGURATION in 1966, the ACM A.M. Turing Award has recognized major contributions of lasting importance to computing. Through the years, it has become the most prestigious award in computing. To help celebrate 50 years of the ACM Turing Award and the visionaries who have received it, ACM has launched a campaign called "Panels in Print," which takes the form of a collection of responses from Turing laureates, ACM award recipients and other ACM experts on a given topic or trend.

For our third Panel in Print, we invited 2009 ACM Prize recipient **ERIC BREWER**, 2004 ACM A.M. Turing Award co-recipient **VINT CERF**, 2016–2017 Athena Lecturer **JENNIFER REXFORD**, ACM Grace Murray Hopper Award recipient **MARTIN CASADO**, ACM Fellows **NICK FEAMSTER** and **JIM KUROSE**, and ACM member **GEORGE ROUSSOS** to discuss the Internet of Things (IoT).

What do you see as some of the biggest transformations that have been brought through the Internet and where do we go next?

ERIC BREWER: The most important transformation brought about by the Internet is the kind of self-empowerment it has caused. If you don't know something, you can find it out. If you want to educate yourself on something, you can learn it fairly directly. People feel like they can learn anything, in any country now.

NICK FEAMSTER: The early Internet was a network of trusted research universities with very few stakeholders. There was no business aspect to it, there were no profits to be taken, and there was little to no concern over security. The chief goal was connectivity, and the primary challenges were technical in nature.

Today, the situation is much different, with each of the previous points having been turned completely on their head. We see increasing tensions between stakeholders, especially between Internet service providers and content providers on to issues like pricing of Internet access, network neutrality, performance guarantees and quality of experience. We also see tremendous tension in cybersecurity between attackers, businesses and end users.

JENNIFER REXFORD: Recently, the Internet has become an amazing way to collect and analyze data about people and their behavior and the kinds of things they do online. This, in turn, has allowed the information we see on the Internet to be much more customized, like Google search and so on. Which brings us to the current evolution, the connecting of the Internet to the physical world, or Internet of Things. This is where we are actually effecting change in the physical world based on the information that gets collected over networks. .

VINT CERF: Projecting into the future, we can see much higher-speed access to the Net, more wireless access and increasing amounts of artificial intelligence and machine learning adding to our ability to accomplish our objectives. It's a rich environment we are heading into.

There are reasons to be concerned—for example, about safety, security, privacy, resilience, and robustness. I am particularly concerned about what I'll call "autonomy," which stems from my concern that you don't want to have a highly automated house that doesn't work when it's not connected to the Internet. So, you need to have local capability independent of or in addition to interactions through the public Internet.

"The most important transformation brought about by the Internet is the kind of self-empowerment it has caused."

There are still more people in the world offline than on. How will connecting these individuals help neglected and underserved communities around the world?

MARTIN CASADO: I agree with the United Nations in the view that connectivity to the Internet is a basic human right. Beyond the intrinsic benefits of better communication within the community, it provides access to the grand marketplace that's erupted within the Internet. In many ways, that can become a great equalizer. If it costs me less to produce a good or a service, and the distribution cost (in this case the Internet) is the same, then I have an advantage in an open market. Of course, it isn't as simple as that, but it certainly does inject underserved communities directly into the economic nervous system in which they can participate.

GEORGE ROUSSOS: The two main factors limiting the ability of people to access the Internet are affordability and lack of literacy and language skills. While getting online can provide benefits, connectivity is not a panacea for all ills. Lifting these communities out of poverty and getting the basics right such as access to clean water, vaccinations, or in some cases a less corrupt government, would be a priority. Moreover, joining the connected world as a latecomer involves significant hazards as well as opportunities, so developing the appropriate skills and safeguards is a precondition.

There are already interesting cases highlighting how innovations can be created from the bottom up: for example, through microlending and using the mobile Internet to broaden access to financial services.

JENNIFER REXFORD: I think there is a lot of opportunity to collect data that can help people make better decisions. For example, farmers could determine the going rate for their crops, rather than relying on a third-party intermediary to determine prices. Knowing what the weather is going to be like in a few days to make decisions about farming practices, and so on. That being said, having access to information for education and training and awareness doesn't replace having access to clean water and very basic needs.

One problem in a lot of the developing world is that much of the Internet traffic is routed back through more developed areas; traffic in South America being routed through Miami, or traffic in Africa going through Amsterdam or London, etc. So there is a missed opportunity to host local content locally. For example, if you're in Kenya, a local Kenyan website will be hosted outside of Kenya, making it very expensive and slow to get information. What we are starting to see more are efforts to have Internet exchange points in the region so that the multiple network providers within Africa and within South America can directly connect with one another and provide a stable platform for hosting of local content.

For organizations and individuals to be confident when conducting transactions and exchanging information, the Internet has to be secure. How does the IoT impact the security of the Internet?

JIM KUROSE: With an ever-increasing array of devices being connected to the Internet (between 26 billion and 50 billion devices in manufacturing, business, and home applications by 2020, by some predictions), the question of resilience—knowing that a device will continue to perform its tasks safely and securely in the presence of unintended as well as malicious faults—is increasingly important.

VINT CERF: There are technologies that allow people to protect themselves better. Two-factor authentications are a good example of that—the best practice of which is to encrypt everything from the laptop or mobile all the way to the server on the net. All of these are practices we adopt at Google.

NICK FEAMTSER: There are a couple of reasons why IoT raises the stakes as far as the security of the Internet is concerned. An Internet attack may now involve physical inconveniences or threats such as security cameras, door locks, thermostats, etc.

The issue here is that most businesses are fundamentally focused on the market they serve. In other words, a hardware company is just a hardware company, a consumer electronics company is just a consumer electronics company. They are not thinking about the security of the software they put on the devices they sell. So it won't be long until we have an abundance of fundamentally unpatchable, insecure, and difficult if not impossible-to-patch devices affecting nearly every aspect of our daily lives. It's a perfect storm.

ERIC BREWER: Even though "less-connected devices" sounds paradoxical in today's scenario, I believe it's an option. As an example, if a device has to connect through the user's phone or home laptop or computer, maybe that is a bit safer because then, at least, the gateway could be secured. Another option is to stop making these devices so flexible. They are really just doing one kind of reporting, and all the rest of the data is in the cloud. It's more plausible that you could make that secure.

What makes security hard is if you are trying to have a lot of flexibility in the device, or complexity, or if you're trying to change what the device is doing over time, and that's why you're having upgrades. All this makes it much more like a phone and then it really needs to have a more automated form of security patching.

What are the possibilities, and repercussions, of IoT capabilities such as smart cities and connected cars?

MARTIN CASADO: There are obvious answers here around energy efficiency, traffic, safety, etc. But I feel that those are already easy to see from where we are today. So perhaps I will take a bit of a longer view and say that in the limit IoT could very well make the notion of a city anachronistic. Cities are largely products of organic growth and physical constraints; close enough for protection and commerce, and far enough away for privacy and access to resources. However, IoT changes these constraints. Drones can deliver goods without requiring traditional roads or supply routes. Advances in connected and urban farming can allow sustainability just about anywhere. And the Internet provides a social overlay that is independent of geography. We are heading toward a future where cities are more defined by common interests than by geography.

What do you think are some of the potentially most exciting/important applications of IoT beyond the ones already being actively developed?

JIM KUROSE: It's difficult to predict future Internet applications. But I'll make *one* prediction. I believe education and skill acquisition have yet to be truly disrupted by the Internet, or by interactive and/or virtual reality/augmented reality technologies. As a long-time teacher (and learner), I don't think there is anything as good as learning with inspired and engaged teachers and students, using interactive learning and team-based activities in the classroom. But that approach is neither uniformly affordable nor scalable. So I do believe a next generation of interactive software/textbooks/classes is increasingly important to meet the pace and need for training, skills updating, and acquiring new fundamentals.

GEORGE ROUSSOS: One specific way that I hope the IoT can bring about change is by shifting the emphasis away from our current predominantly visual mode of interaction with information, which I consider to be the key ingredient enabling a sedentary and passive contemporary lifestyle. IoT technologies afford interactions engaging the whole body through touch, proprioception, equilibrioception, interoception, and perhaps a few new artificial senses that can hopefully rebalance the focus on the brain as the only locus of intelligence.

In particular, my hope is that the IoT will play a key role toward improving the health and the sustainability of the planet: overconsumption of raw materials, pollution from fossil fuels, and industrialized farming, the destruction of forests and numerous other effects of modernity are setting massive challenges ahead. I believe the IoT has to play a central role in addressing these challenges and ensuring the welfare of future generations. **C**

CREDIT TK

ACM
ON A MISSION TO SOLVE TOMORROW.

Dear Colleague,

Without computing professionals like you, the world might not know the modern operating system, digital cryptography, or smartphone technology to name an obvious few.

For over 60 years, ACM has helped computing professionals be their most creative, connect to peers, and see what's next, and inspired them to advance the profession and make a positive impact.

We believe in constantly redefining what computing can and should do.

ACM offers the resources, access and tools to invent the future. No one has a larger global network of professional peers. No one has more exclusive content. No one presents more forward-looking events. Or confers more prestigious awards. Or provides a more comprehensive learning center.

Here are just some of the ways ACM Membership will support your professional growth and keep you informed of emerging trends and technologies:

- Subscription to ACM's flagship publication ***Communications of the ACM***
- Online books, courses, and videos through the **ACM Learning Center**
- Discounts on registration fees to ACM Special Interest Group conferences
- Subscription savings on specialty magazines and research journals
- The opportunity to subscribe to the **ACM Digital Library**, the world's largest and most respected computing resource

Joining ACM means you dare to be the best computing professional you can be. It means you believe in advancing the computing profession as a force for good. And it means joining your peers in your commitment to solving tomorrow's challenges.

Sincerely,

Vicki L. Hanson
President
Association for Computing Machinery

acm
Association for
Computing Machinery
Advancing Computing as a Science & Profession

DOI:10.1145/3068787

Woodrow Hartzog and Ira Rubinstein

Law and Technology

The Anonymization Debate Should Be About Risk, Not Perfection

Focusing on the process of anonymity rather than pursuing the unattainable goal of guaranteed safety.

For years, the key ethic for safe, sustainable data sharing was anonymization. As long as a researcher or organization took steps to anonymize datasets, they could be freely used and shared. This notion was even embedded in law and policy. For example, laws like the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule and the European Union's Data Protection Directive facilitate sharing of anonymized datasets with fewer if any restrictions placed upon datasets that contain personal information.

But it turns out that "anonymization" is not foolproof. The possibility of correctly identifying people and attributes from anonymized datasets has sparked one of the most lively and important debates in privacy law. In the past 20 years, researchers have shown that individuals can be identified in many different datasets once thought to have been fully protected by means of de-identification.[a,7] In particular, a trio of well-known cases of re-identification has called into question the validity of the de-identification methods on which privacy law and policy, like the HIPAA privacy rule, relies. A governor and Netflix and AOL customers were all accurately identified from purportedly anonymized data. In each case, an adversary took advantage of auxiliary information to link an individual to a record in the de-identified dataset.

The failure of anonymization has been widely publicized. But the debate over how to proceed in policy and practice remains stalled. In order to find the right path, the perfect cannot be the enemy of the good. Anonymization must be conceptualized as a process of minimizing risk instead of a state of guaranteed safety.

A Crisis of Faith and Scientific Discord

The possibility of correctly identifying people and attributes from de-identified datasets has sparked a crisis of faith in the validity of de-identification methods. Do these methods still protect data subjects against possible privacy harms associated with revealing sensitive and non-public information? Certainly, there is widespread skepticism about de-identification techniques among some leading privacy scholars and most of the popular press, which in turn undermines the credibility of the exemptions for de-identified data in regimes like HIPAA. This is of obvious concern because it not only creates legal and regulatory uncertainty for the scientific research community but may even discourage individuals from contributing data to new research

[a] Paul Ohm, *Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization.* 57 *UCLA Law Review* 1701, 2010. Ohm introduced the legal community to the relevant computer science literature, including a classic attack on the Netflix Prize dataset; see Narayanan and Shmatikov.[7]

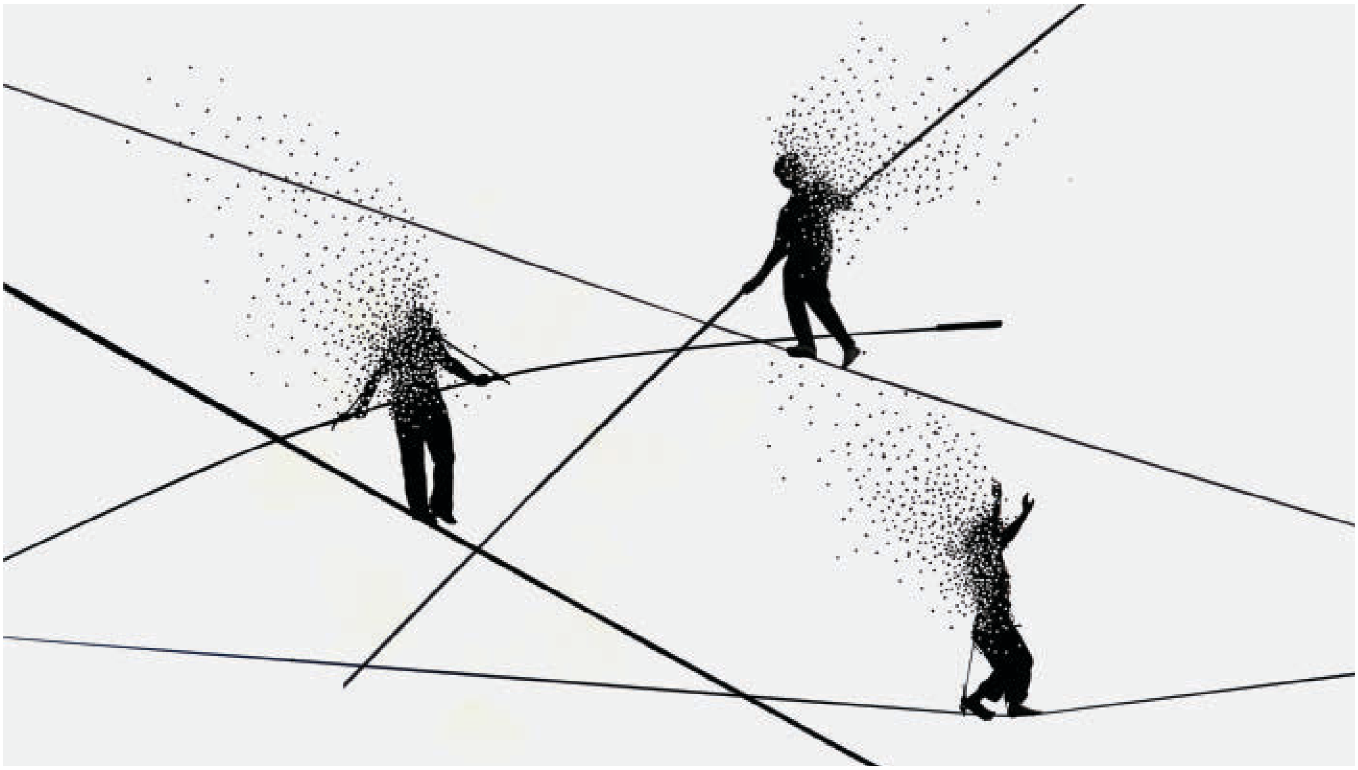


IMAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

projects. (It also heightens consumer mistrust of e-commerce firms offering their own dubious "guarantees" of anonymization, thereby reinforcing the "privacy is dead" meme.)

The community of computer scientists, statisticians, and epidemiologists who write about de-identification and re-identification are deeply divided, not only in how they view the implications of the auxiliary information problem, but in their goals, methods, interests, and measures of success. Indeed, we have found that the experts fall into two distinct camps. First, there are those who may be categorized as "pragmatists" based on their familiarity with and everyday use of de-identification methods and the value they place on practical solutions for sharing useful data to advance the public good.[1] Second, there are those who might be called "formalists" because of their insistence on mathematical rigor in defining privacy, modeling adversaries, and quantifying the probability of re-identification.[6] Pragmatists devote a great deal of effort to devising methods for measuring and managing the risk of re-identification for clinical trials and other specific disclosure scenarios. Unlike their formalist adversaries, they consider it difficult to gain access to auxiliary information and consequently give little weight to attacks demonstrating that data subjects are distinguishable and unique but that (mostly) fail to re-identify anyone on an individual basis. Rather, they argue that empirical studies and meta-analyses show that the risk of re-identification in properly de-identified datasets is, in fact, very low.

Formalists, on the other hand, argue that efforts to quantify the efficacy of de-identification "are unscientific and promote a false sense of security by assuming unrealistic, artificially constrained models of what an adversary might do."[6] Unlike the pragmatists, they take very seriously proof-of-concept demonstrations of re-identification, while minimizing the importance of empirical studies showing low rates of re-identification in practice.

This split among the experts is concerning for several reasons. Pragmatists and formalists represent distinctive disciplines with very different histories, questions, methods, and objectives. Accordingly, they have shown little inclination to engage in fruitful dialogue much less to join together and find ways to resolve their differences or place de-identification on firmer foundations that would eliminate or at least reduce the skepticism and uncertainty that currently surrounds it. And this makes it very difficult for policy makers to judge whether the HIPAA de-identification rules should be maintained, reformed, or abandoned.

These divergent views might lead us to different regulatory approaches. Those that focus on the remote possibility of re-identification might prefer an approach that reserves punishment only in the rare instance of harm, such as a negligence or strict liability regime revolving around harm triggers. Critics of anonymization might suggest we abandon de-identification-based approaches altogether, in favor of different privacy protections focused on collection, use, and disclosure that draw from the Fair Information Practice Principles, often called the FIPPs.

These problems with the de-identification debate are frustrating sound data use policy. But there is a way forward. Regulators should incorporate the full gamut of Statistical Disclosure Limitation (SDL) methods and techniques into privacy law and policy, rather than relying almost exclusively on de-identification techniques that only modify and obfuscate data. SDL comprises the principles and techniques that researchers have developed for disseminating official statistics and other data for research purposes while protecting the privacy and confidentiality of data subjects. SDL can be thought of in terms of three major forms of interaction between researchers and personal data: direct access (which covers access to data by qualified investigators who must agree to licensing terms and access datasets securely); dissemination-based access (which includes de-identification), and query-based access (which includes but is not limited to differential privacy).[5]

Adopting the SDL frame for the de-identification debate helps to clarify several contested issues in the current debate. First, the most urgent need today is not for improved de-identification methods alone but also for research that provides agencies with methods and tools for making sound decisions about SDL. Second, the SDL literature calls attention to the fact that researchers in statistics and computer science pursue very different approaches to confidentiality and privacy and all too often do so in isolation from one another. They might achieve better results by collaborating across methodological divides. Third, the legal scholars who have written most forcefully on this topic tend to evaluate the pros and cons of de-identification in isolation from other SDL methods. Debates focusing exclusively on the merits or demerits of de-identification are incomplete. SDL techniques should be part of most regulators' toolkits.

The Way Forward: Minimizing Risk

Most importantly, SDL can be leveraged to move de-identification policy toward a process of minimizing risk. A risk-based approach would seek to tailor SDL techniques and related legal mechanisms to an organization's anticipated privacy risks. For example, if the federal agency administering the HIPAA Privacy Rule (Health and Human Services) fully embraced a risk-based approach, this would transform the rule into something more closely resembling the law of data security.[4] Such an approach would have three major features:

Process-based: Organizations engaged in releasing data to internal, trusted, or external recipients should assume responsibility for protecting data subjects against privacy harms by imposing technical restrictions on access, using adequate de-identification procedures, and/or relying on query-based methods, all in combination with legal mechanisms, as appropriate.

Statistical Disclosure Limitation can be leveraged to move de-identification policy toward a process of minimizing risk.

Contextual: Sound methods for protecting released datasets are always contingent upon the specific scenario of the data release. There are at least seven variables to consider in any given context, many of which have been previously identified in reports by the National Institute of Standards and Technology (NIST) and others. They include data volume, data sensitivity, type of data recipient, data use, data treatment technique, data access controls, and consent and consumer expectations.

Tolerant of risk: The field of data security has long acknowledged there is no such thing as perfect security. If the Weld, AOL, and Netflix re-identification incidents prove anything, it is that perfect anonymization also is a myth. By focusing on process instead of output, data release policy can aim to raise the cost of re-identification and sensitive attribute disclosure to acceptable levels without having to ensure perfect anonymization.[b]

Organizations sharing data should be required to provide "reasonable data release protections." The tenets of reasonable, process-based, data-release protections would look similar to those of data security: assess data to be shared and risk of disclosure; minimize data to be released; implement reasonable de-identification and/or additional data control techniques as appropriate; and develop a monitoring, accountability, and breach response plan.

These requirements would be informed by the nascent industry standards under development by NIST and others, including accepted de-identification and SDL techniques as well as a consideration of the risk vectors described here.[2] Of course, those who engage in unauthorized re-identification are also culpable and it might be worthwhile to supplement contractual or statutory obligations not to engage in re-identification with severe civil (or even criminal) penalties for intentional violations that cause harm.[3] It is important that any such statutory prohibitions also include robust exemptions for security research into de-identification and related topics.

A risk-based approach recognizes there is no perfect anonymity. It focuses on process rather than output. Yet effective risk-based data release policy also avoids a ruthless pragmatism by acknowledging the limits of current risk projection models and building in important protections for individual privacy. This policy-driven, integrated, and comprehensive approach will help us better protect data while preserving its utility.

[b] This Viewpoint is based on a longer article by the co-authors, which provides a more detailed discussion of these three factors; see Rubinstein, I. and Hartzog, W. *Anonymization and Risk. 91 Washington Law Review* 703, 2016.

References

1. Cavoukian, A. and El Emam, K. *Dispelling the Myths Surrounding Deidentification: Anonymization Remains a Strong Tool for Protecting Privacy.* Information and Privacy Commissioner of Ontario, 2011; http://bit.ly/2nJEcNn
2. Garfinkel, S.L. *De-Identification of Personal Information.* National Institute of Standards and Technology, 2015; http://bit.ly/2cz28ge
3. Gellman, R. The deidentification dilemma: A legislative and contractual proposal. 21 *Fordham Intell. Prop. Media & Ent. L.J.* 33, 2010.
4. Hartzog, W. and Solove, D.J. The scope and potential of FTC data protection. 83 *Geo. Washington Law Review 2230*, 2015.
5. Kinney, S.K. et al. Data confidentiality: The next five years summary and guide to papers. *J. Privacy and Confidentiality 125* (2009).
6. Narayanan, A. and Felten, E.W. No silver bullet: De-identification still doesn't work, 2014; http://bit.ly/1kEPwxV
7. Narayanan, A. and Shmatikov, V. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 29th IEEE Symposium on Security and Privacy* 111.

Woodrow Hartzog (whartzog@samford.edu) is a Starnes Professor of Law with the Cumberland School of Law at Samford University.

Ira Rubinstein (ira.rubinstein@nyu.edu) is a Senior Fellow at the Information Law Institute at New York University School of Law.

DOI:10.1145/3068791

Leo Porter, Cynthia Lee, Beth Simon, and Mark Guzdial

Education

Preparing Tomorrow's Faculty to Address Challenges in Teaching Computer Science

Using a "boot camp" workshop for new faculty orientation.

IMAGE FROM SHUTTERSTOCK.COM

As computing becomes more pervasive, we see increased demand from students eager to start a career in computing, and also from students in related disciplines recognizing the need for computer science skills. The result is increased overall enrollments—in some schools, by a factor of three in the past five years.[a,b] Higher enrollment leads to ballooning class sizes. Schools struggle to hire and retain faculty in the face of heavy courting by industry. The result is that a sense of resource scarcity dominates the high-pressure environment of large class sizes.

The new challenges compound existing teaching-related challenges for the field. We still need to broaden participation in our field, with the lowest percentage of women majors in all of STEM.[c] The economic rewards of a computing career make it even more important to bridge the digital divide. If there are more students than faculty can teach effectively, they may be inclined to lean on a pessimistic belief that success is dependent on "brilliance" and innate ability where only a subset of students can succeed. If CS faculty feel there is little they can do to change students' outcomes in their individual classrooms, it will be true. Research shows that more CS faculty hold this mistaken and unproductive view of students than faculty in other STEM disciplines.[3]

The truth offers more optimism than this. We have a body of evidence supporting specific, replicable practices for creating agile, student-responsive classrooms that provably scale to large classes, reduce failure rates, and broaden participation. Broad awareness and adoption of these evidence-based teaching practices can help address our teaching challenges and reinvigorate

[a] http://www.geekwire.com/2014/analysis-examining-computer-science-education-explosion/

[b] http://cra.org/wp-content/uploads/2016/07/BoomCamp.pdf

[c] https://ngcproject.org/statistics

faculty enjoyment of teaching, even in these outsized conditions. To achieve this goal of disseminating effective teaching methods, we are implementing a strategy that has been successful in other academic disciplines—putting newly hired faculty through a CS New Faculty Teaching Workshop,[d] a rigorous "boot camp" workshop on how to be effective teachers.

Many universities offer some kind of orientation to teaching for new faculty. We are taking a page from other STEM disciplines in teaching a CS-specific introduction for new faculty at research institutions. With the credibility that can only be obtained through being practicing CS faculty ourselves, attendees at the CS New Faculty Teaching Workshop learn methods that have been shown to work in CS classes. Examples are drawn from the specific CS courses and topics the attendees will teach in the coming year, using tools that are specific to the needs of CS homework and programming projects. We teach CS faculty how to succeed as CS faculty, without spending time adapting more general or maybe even inapplicable teaching strategies.

Addressing Challenges

To address the challenges in teaching computer science, our CS New Faculty Teaching Workshop has two long-term aspirations:

- Change practices in the classroom to be evidence-based, for the benefit of faculty and students.
- Change faculty perceptions of teaching so they view it as a scholarly endeavor, leveraging their scientific thinking to continually grow and improve as instructors.

Adoption of evidence-based practices at scale in CS classes could have profound outcomes. Other STEM disciplines are reaping the benefits of active learning. STEM students are learning more and failing less in active learning classes compared to traditional lectures.[1] Evidence-based teaching practices in CS classes leads to better performance on final exams[5] and increased retention of majors.[4] The most common teaching practice in CS remains an apprenticeship model—we lecture and expect students to figure out things on their own. We need broad-based adoption of active learning for the sake of our students.

The second aspiration is to change how CS faculty view education and teaching. CS faculty frequently express beliefs not only that programming is an innate ability, but that good CS educators are "born, not made." There is no incentive for CS faculty to improve, to learn to become better educators, if they think their teaching ability is preordained. And yet, there are many techniques shown by the CS education literature to improve teaching and student outcomes. What we need is for faculty to both value education as part of their profession (we believe most do already!) and for them to leverage the CS education research literature as a source of vetted ideas and ready solutions.

Strategy: Teach the New Faculty to Influence the Future

New CS faculty going into research-intensive universities rarely have much teaching experience. They are hired because of their excellence in research and innovation. CS New Faculty Teaching Workshop attendees often have some anxiety about teaching. They want to do well at it but recognize the challenges they face in large classrooms with students from diverse cultural and programming backgrounds. These are exactly the faculty whom we want to develop and support. We want to give them the tools to be successful in the classroom; to be effective and efficient so they can succeed at both teaching and research. Because they have a strong desire to do right by their students, they are our starting point for developing a culture that values teaching and computing education research.

New CS faculty going into research-intensive universities rarely have much teaching experience.

For the CS New Faculty Teaching Workshop to have the scale of impact we envision, we had to think carefully about strategic audience targeting. We ultimately want to help teaching across all computer science undergraduate departments address today's challenges. But we are starting with the research-intensive universities because they serve as models nationally. The top research institutions simply have inordinate influence in the rest of the computer science education ecosystem. Literally, a handful of schools produce most of the Ph.D.'s who go on to be computer science faculty in the U.S.[e] These newly hired faculty will also go on to hold positions of leadership and influence within their departments and schools, magnifying the impact of the cohort we directly touch each year. As we influence enough faculty that they form a critical mass at each of their home departments, they will have local peer support. With an orientation toward a scholarly view of teaching, we equip them with today's known best practices, and with an inclination to follow future scholarly advances in teaching for tomorrow's challenges.

Changing ingrained teaching practices can be difficult. After all, we succeeded through years of our own student experiences, so we see ourselves as experts. This model of offering new research faculty workshops has been effective in other disciplines. The Physics and Astronomy New Faculty Workshop has been successful in effecting change by reaching approximately 25% of new hires—and at least half of those who attended the workshop reporting adoption of evidence-based practices in their teaching after the workshop.[2] Moreover, participants and department chairs reported a change in culture based on discussions about teaching. Charles Henderson, who evaluated the workshop, suggests it was successful because it targeted new faculty in only a single discipline and presented a wide variety of pedagogical options for potential adoption. Everyone likes to make choices, and if you know more than one way to teach something, you get to make choices and improve your enjoyment of teaching.

[d] http://bit.ly/New-CS-FTW

[e] https://www.microsoft.com/en-us/research/blog/2015-faculty-summit-informs-and-inspires/

Workshop Content and Focus

Emphasize best practices underrepresented groups. New research faculty want, and need, to hear about the importance of evidence-based teaching practices from respected research faculty, not just from a bunch of education researchers (us). We were delighted that a well-recognized researcher and leader, Ed Lazowska, was willing to give the keynote address to kick off the workshop. Lazowska is the Bill and Melinda Gates Chair in Computer Science and Engineering at the University of Washington. At each workshop, his keynote perfectly articulated his own deep passion for teaching and the case for caring about teaching, striving to improve teaching through practice and from the education literature, and for balancing teaching with other faculty responsibilities.

The workshop included sessions interleaving standard know-how and practical starting materials with more advanced methods and evidence from the research literature. And as workshop organizers, we practice what we preach—not just lecturing but involving participants with a variety of engagement techniques. One particularly lively activity involved faculty attempting to rank various pieces of career advice participants might hear at the water cooler as "definitely a good idea," "sometimes a good idea," or "not a good idea."

Topics discussed included student-centered teaching, syllabus development, academic integrity prevention and response, TA management, essential tools for teaching at scale, creating an inclusive classroom, scientific-minded teaching, peer instruction, and other forms of active learning, exam-writing strategies, creating videos and other online content, and how to balance teaching and research responsibilities. Most critically for addressing the challenges of today's computing classrooms, the pedagogies taught are able to scale to large classrooms and the tips for creating an inclusive classroom can broaden participation.

Early Results

We have been very pleased with the success of the two annual CS New Faculty Teaching Workshops we have run so far: A pilot year with eight attendees, scaled up to 22 attendees in the second year. A critical metric of success was the increasing number of applications for our second workshop. This tells us that department chairs at our targeted research-focused institutions are not only getting the word out to their new faculty, but are communicating the value of the CS New Faculty Workshop to junior faculty members.

A critical metric of success was the increasing number of applications for our second workshop.

Additionally, survey highlights from participants in the 2016 workshop indicate:

- 89% found the workshop to be "very valuable" (with the remaining 11% indicating it was minimally or moderately valuable);
- The most highly rated session was on "Research on Active Learning" with 95% indicating it was moderately or very valuable; and
- 100% said the workshop techniques would help improve their teaching.

The most commonly mentioned take-away from the workshop was that participants planned to implement some form of active learning in the classroom. We were pleasantly surprised that in such a short time so many participants came to express the importance of working toward engaging students with more active learning in their classrooms. Consider that participants had just eight or more years of higher education where they likely never saw active learning modeled for them in any of their classes. After less than two days of exposure, they were convinced of the importance, and felt they were given enough concrete guidance and examples that they could start using active learning techniques in their classrooms. Follow-up evaluation on each cohort is being conducted to see how many have put the techniques to use in their classes in the year following their attendance.

Most critically to the future of the program, one participant said: "I'm going to start to recommending this workshop to all new faculty." We will only succeed in making our desired cultural changes if we can draw in a critical mass of new faculty at these institutions. If a second-year professor tells a new hire or a former graduate student colleague heading off to the professoriate, "This is worth your time, you should go," we will have our most effective recruitment device.

Creating a Cohort for Change

Real change takes time. If our workshops continue to be successful, we will see change coming as new faculty advance and share their new perspective on teaching with their colleagues. As the CS New Faculty Teaching Workshop continues, our CS faculty will be adept at facing challenges in teaching by having adopted evidence-based teaching practices and by having a scholarly attitude about teaching. C

References

1. Freeman, S. et al. Active learning increases student performance in science, engineering, and mathematics. In *Proceedings of the National Academy of Sciences 111 23* (2014), 8410–8415.
2. Henderson, C. Promoting instructional change in new faculty: An evaluation of the Physics and Astronomy new faculty workshop. *American Journal of Physics 76*, 2 (2008), 179–187.
3. Leslie, S.-J. et al. Expectations of brilliance underlie gender distributions across academic disciplines. *Science 347*, 6219 (2015), 262–265.
4. Porter, L. and Simon, B. Retaining nearly one-third more majors with a trio of instructional best practices in CS1. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, ACM 2013, 165–170.
5. Simon, B. et al. How we teach impacts student learning: Peer instruction vs. lecture in CS0. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, ACM 2013, 41–46.

Leo Porter (leporter@eng.ucsd.edu) is an Assistant Teaching Professor in the Department of Computer Science and Engineering at the University of California, San Diego.

Cynthia Lee (cbl@stanford.edu) is a Lecturer in the Computer Science Department at Stanford University.

Beth Simon (bsimon@ucsd.edu) is an Associate Teaching Professor in Education Studies at the University of California, San Diego.

Mark Guzdial (guzdial@cc.gatech.edu) is a Professor in the School of Interactive Computing at Georgia Institute of Technology.

The CS New Faculty Teaching Workshop is supported through a grant from the National Science Foundation DUE-1432830.

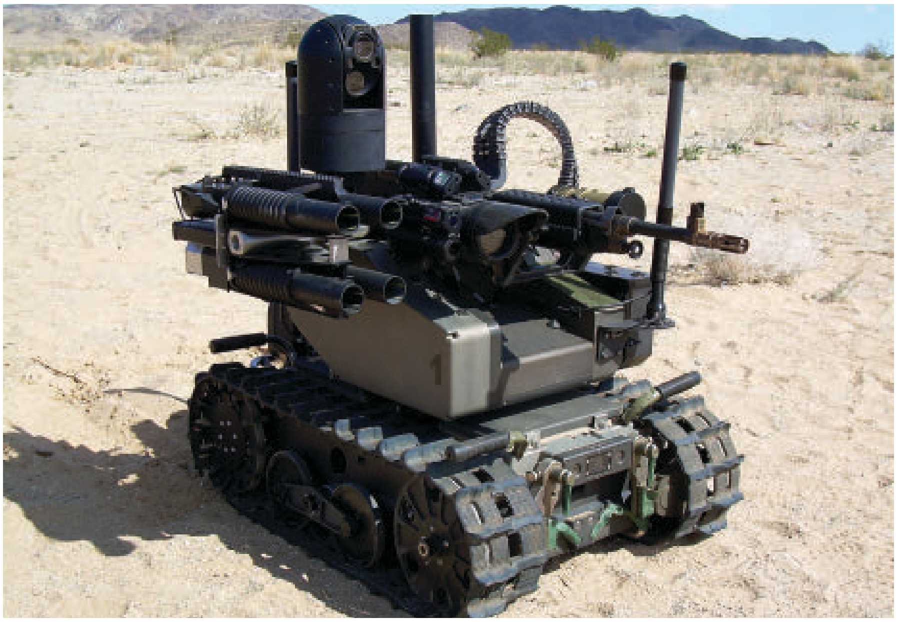
DOI:10.1145/2998579

Wendell Wallach

Viewpoint
Toward a Ban on Lethal Autonomous Weapons: Surmounting the Obstacles

A 10-point plan toward fashioning a proposal to ban some—if not all—lethal autonomous weapons.

FROM APRIL 11–15, 2016, at the United Nations Office at Geneva, the Convention on Certain Conventional Weapons (CCW) conducted a third year of informal meetings to hear expert testimony regarding a preemptive ban on lethal autonomous weapons systems (LAWS). A total of 94 states attended the meeting, and at the end of the week they agreed by consensus to recommend the formation of an open-ended Group of Government Experts (GGE). A GGE is the next step in forging a concrete proposal upon which the member states could vote. By the end of 2016 a preemptive ban has been called for by 19 states. Furthermore, *meaningful human control*, a phrase first proposed by advocates for a ban, has been adopted by nearly all the states, although the phrase's meaning is contested. Thus a ban on LAWS would appear to have gained momentum. Even the large military powers, notably the U.S., have publicly stated that they will support a ban if that is the will of the member states. Behind the scenes, however, the principal powers express their serious disinclination to embrace a ban. Many of the smaller states will follow their lead. The hurdles in the way of a successful campaign to ban LAWS remain daunting, but are not insurmountable.



The Modular Advanced Armed Robotic System is an unmanned ground vehicle for reconnaissance, surveillance, and target acquisition missions.

IMAGE COURTESY OF QINETIQ NORTH AMERICA

The debate to date has been characterized by a succession of arguments and counterarguments by proponents and opponents of a ban. This back and forth should not be interpreted as either a stalemate or a simple calculation as to whether the harms of LAWS can be offset by their benefits. For all states that are signatories to the laws of armed conflict,[a] any violation of the principles of international humanitarian law (IHL)[b] must trump utilitarian calculations. Therefore, those who believe the benefits of LAWS justify their use and therefore oppose a ban, are intent that LAWS do not become a special case within IHL. Demonstrating that LAWS pose unique challenges

[a] LOAC, also known as International Humanitarian Law (IHL), is codified in the Geneva Conventions and additional Protocols. The laws seek to limit the effects of armed conflict, particularly the protection of non-combatants.

[b] Four principles of IHL provide protection for civilians: distinction, necessity and proportionality, humane treatment, and nondiscrimination.

for IHL has been a core strategy for supporters of a ban.

Those among the more than 3,100 AI/Robotics researchers who signed the *Autonomous Weapons: An Open Letter From AI & Robotics Researchers*[c] are reflective of a broad consensus among citizens and even active military personnel who favor a preemptive ban.[4] This consensus is partially attributable to speculative, futuristic, and fictional scenarios. But perhaps even science fiction represents a deep intuition that unleashing LAWS is not a road humanity should tread.

Researchers who have waded into the debate over banning LAWS have come to appreciate the manner in which geopolitics, security concerns, the arcana of arms control, and linguistic obfuscations can turn a relatively straightforward proposal into an extremely complicated proposition. A ban on LAWS does not fit easily, or perhaps at all, into traditional models for arms control. If a ban, or even a moratorium, on the development of LAWS is to progress, it must be approached creatively.

I favor and have been a long-time supporter of a ban. While a review of the extensive debate as to whether LAWS should be banned is well beyond the scope of this paper, I wish to share a few creative proposals that could move the campaign to ban LAWS forward. Many of these proposals were expressed during my testimony at the CCW meeting in April and during a side luncheon event.[d] Before introducing those proposals, let me first point out some of the obstacles to fashioning an arms control agreement for LAWS.

Why Banning LAWS Is Problematic

▸ Unlike most other weapons that have been banned, some uses of LAWS are perceived as morally acceptable, if not morally obligatory. The simple fact that LAWS can be substituted for and thus save the lives of one's own soldiers is the most obvious moral good. Unfortunately, this same moral good lowers the barriers to initiating new wars. Some nations will be emboldened to start wars if they believe they can achieve political objectives without the loss of their troops.

Some nations will be emboldened to start wars if they believe they can achieve political objectives without the loss of their troops.

▸ It is unclear whether armed military robots should be viewed as weapon systems or weapon platforms, a distinction that has been central to many traditional arms control treaties. Range, payload, and other features are commonly used in arms control agreements to restrict the capabilities of a weapon system. A weapon platform can be regulated by restricting where it can be located. For example, agreements to restrict nuclear weapons will specify number of warheads and the range of the missiles upon which they are mounted, and even where the missiles can be stationed. With LAWS, what is actually being banned?

• Arms control agreements often focus on working out modes of verification and inspection regimes to determine whether adversaries are honoring the ban. The difference between a lethal and non-lethal robotic system may be little more than a few lines of code or a switch, which would be difficult to detect and could be removed before or added after an inspection. Proposed verification regimes for LAWS[6] would be extremely difficult and costly to enforce. Military strategists do not want to restrict their options, when that of bad actors is unrestricted.

• LAWS differ in kind from the various weapon systems that have to date been banned without requiring an inspection regime. Consider, for example, the relatively recent bans on blinding lasers or anti-personnel weapons, which are often offered as a model for arms control for LAWS. These bans rely on representatives of *civil society*, non-governmental organizations such as the International Committee of the Red Cross, to monitor and stigmatize violations. So also will a ban on LAWS. However, blinding lasers and anti-personnel weapons were relatively easy to define. After the fact, the use of such weapons can be proven in a straightforward manner. Lethal autonomy, on the other hand, is not a weapon system. It is a feature set that can be added to many, if not all, weapon systems. Furthermore, the uses of autonomous killing features are likely to be masked.

• LAWS will be relatively easy to assemble using technologies developed for civilian applications. Thus their proliferation and availability to non-state actors cannot be effectively stopped.

In forging arms-control agreements definitional distinctions have always been important. Contentions that definitional consensus cannot be reached for *autonomy* or *meaningful human control*, that LAWS depend upon advanced AI, and that such systems are merely a distant speculative possibility repeatedly arose during the April discussion at the U.N. in Geneva, and generally served to obfuscate, not clarify, the debate. A circular and particularly unhelpful debate has ensued over the meaning of *autonomy*, with proponents and opponents of a ban struggling to establish a definition that serves their cause. For example, the U.K. delegation insists that *autonomy* implies near humanlike capabilities[e] and anything short of this is merely an *automated* weapon. The Campaign to Stop Killer Robots favors a definition where *autonomy* is the ability to perform a task without immediate intervention from a human. Similarly, definitions for *meaningful human control* range from a military leader specifying a kill order in advance of deploying a weapon system to having the real-time engagement of a human *in the loop* of selecting and killing a human target.

[c] Available at http://bit.ly/1V9bls5

[d] The full April 12, 2016, testimony entitled, Predictability and Lethal Autonomous Weapons Systems (LAWS), is available at http://bit.ly/2mjmuwH. An extended article accompanied this testimony. That article was circulated to all the CCW member states by the chair of the meeting, Ambassador Michael Biontino of Germany. It was also published in Robin Geiss, Ed., 2017, "Lethal Autonomous Weapons Systems: Technology, Definition, Ethics, Law & Security." Federal Foreign Office, p. 295–312. The luncheon event on April 11, 2016, was sponsored by the United Nations Institute for Disarmament Research (UNIDIR).

[e] While the U.K. representatives did not use this language, it does succinctly capture the delegation's statements that all computerized systems are merely automated until they display advanced capabilities.

The leading military powers contend that they will maintain effective control over the LAWS they deploy.[f] But even if we accept their sincerity, this totally misses the point. They have no means of ensuring that other states and non-state actors will follow suit.

More is at stake in these definitional debates than whether to preemptively ban LAWS. Consider a Boston Dynamic's Big Dog loaded with explosives, and directed through the use of a GPS to a specific location, where it is programmed to explode. Unfortunately, during the time it takes to travel to that location, the site is transformed from a military outpost to a makeshift hospital for injured civilians. A strong definition for *meaningful human control* would require the location be given a last-minute inspection before the explosives could detonate. Big Dog, in this example, is a dumb LAW, which we should perhaps fear as much as speculative future systems with advanced intelligence. Dumb LAWS, however, do open up comparisons to widely deployed existing weapon systems, such as cruise missiles, whose impact on an intended target military leaders have little or no ability to alter once the missile has been launched. In other words, banning dumb LAWS quickly converges with other arms control campaigns, such as those directed at limiting cruise missiles and ballistic missiles.[5] States will demand a definition for LAWS that distinguishes them from existing weapon systems.

Delegates at the CCW are cognizant that in the past (1990s) they failed at banning the dumbest, most indiscriminate, and autonomous weapons of all, anti-personnel mines. Nevertheless, anti-personnel weapons (land mines) were eventually banned during an independent process that led up to the Mine Ban or Ottawa Treaty; 162 countries have committed to fully comply with that treaty.[g]

The leading military powers contend they will maintain effective control over the LAWS they deploy. But even if we accept their sincerity, this totally misses the point.

A second failure to pass restrictions on the use of a weapon systems, whose ban has garnered popular support, might damage the whole CCW approach to arms control. This knowledge offers the supporters of a ban a degree of leverage presuming: the ban truly has broad and effective public support; LAWS can be distinguished from existing weaponry that is widely deployed; and creative means can be forged to develop the framework for an agreement.

A 10-Point Plan

Many of the barriers to fitting a ban on LAWS into traditional approaches to arms control can be overcome by adopting the following approach.

1. Rather than focus on establishing a bright line or clear definition for *lethal autonomy*, first establish a high order moral principle that can garner broad support. My candidate for that principle is: *Machines, even semi-intelligent machines, should not be making life and death decisions. Only moral agents should make life and death decisions about humans.* Arguably, something like this principle is already implicit, but not explicit, in existing international humanitarian law, also known as the laws of armed conflict (LOAC).[3] A higher order moral principle makes explicitly clear what is off limits, while leaving open the discussion of marginal cases where a weapon system may or may not be considered to be *making life and death decisions*.

2. Insist that *meaningful human control* and *making a life and death decision* requires the real-time authorization from designated military personnel for a LAW to kill a combatant or destroy a target that might harbor combatants and non-combatants alike. In other words, it is not sufficient for military personnel to merely delegate a kill order in advance to an autonomous weapon or merely be "on-the-loop"[h] of systems that can act without a real time go-ahead.

3. Petition leaders of states to declare that LAWS violate existing IHL. In the U.S. this would entail a Presidential Order to that effect.[i,14]

4. Review marginal or ambiguous cases to set guidelines for when a weapon system is truly autonomous and when its actions are clearly the extension of a military commander's will and intention. Recognize that any definition of autonomy will leave some cases ambiguous.

5. Underscore that some present and future weapon system will occasionally act unpredictably and most LAWS will be difficult if not impossible to test adequately.

6. Present compelling cases for banning at least some, if not all, LAWS. In other words, highlight situations in which nearly all parties will support a ban. For example, no nation should want LAWS that can launch nuclear warheads.

7. Accommodate the fact that there will be necessary exceptions to any ban. For example, defensive autonomous weapons that target unmanned incoming missiles are already widely deployed.[j] These include the U.S. Aegis Ballistic Missile Defense System and Israel's Iron Dome.

8. Recognize that future technological advances may justify additional

[f] See, for example, the U.S. Department of Defense Directive 2000.09 entitled, "Autonomy in Weapon Systems." The Directive is dated November 21, 2012 and signed by Deputy Secretary of Defense, Ashton B. Carter, who was appointed Secretary of Defense by President Obama on December 5, 2014; http://bit.ly/1myJikF

[g] The U.S., Russia, and China are not signatories to the Ottawa Treaty, although the U.S. has pledged to largely abide by its terms.

[h] "On the loop" is a term that first appeared in the "United States Air Force Unmanned Aircraft Systems Flight Plan 2009–2047." The plan states: Increasingly humans will no longer be "in the loop" but rather "on the loop"—monitoring the execution of certain decisions. Simultaneously, advances in AI will enable systems to make combat decisions and act within legal and policy constraints without necessarily requiring human input.

[i] Wallach, W. (2012, unpublished but widely circulated proposal). Establishing limits on autonomous weapons capable of initiating lethal force.

[j] In practice a weapon designed for defensive purposes might be used offensively. So the distinction between the two should emphasize the use of defensive weaponry to target unmanned incoming missiles.

exceptions to a ban. Probably the use of LAWS to protect refugee non-combatants would be embraced as an exception. Whether the use of LAWS in a combat zone where there are no non-combatants should be treated as an exception to a ban would need to be debated. Offensive autonomous weapon systems that do not target humans, but only target, for example, unmanned submarines, might be deemed an exception.

9. Utilize the unacceptable LAWS to campaign for a broad ban, and a mechanism for adding future exceptions.

10. Demand that the onus of ensuring that LAWS will be controllable, and that those who deploy the LAWS will be held accountable, lies with those parties who petition for, and deploy, an exception to the ban.

Unpredictable Behavior: Why Some LAWS Must Be Banned

A ban will not succeed unless there is a compelling argument for restricting at least some, if not all, LAWS. In addition to the ethical arguments for and against LAWS, concern has been expressed that autonomous weapons will occasionally behave unpredictably and therefore might violate IHL, even when this is not the intention of those who deploy the system. The ethical arguments against LAWS have already received serious attention over the past years and in the ACM. During my testimony at the CCW in April 2016, I fleshed out why the prospect of unanticipated behavior should be taken seriously by member states. The points I made are fairly well understood within the community of AI and robotics' engineers, and go beyond weaponry to our ability to predict, test, verify, validate, and ensure the behavior and reliability of software and indeed any complex system. In addition, debugging and ensuring that software is secure can be a costly and a never-ending challenge.

Factors that influence a system's predictability. Predictability for weaponry means that within the task limits for which the system is designed, the anticipated behavior will be realized, yielding the intended result. However, nothing less than a law of physics is absolutely predictable. There are only degrees of predictability, which in theory can be represented as a probability. Many factors influence the predictability of a system's behavior, and whether operators can properly anticipate the system's behavior.

- An unanticipated event, force, or resistance can alter the behavior of even highly predictable systems.
- Many if not most autonomous systems are best understood as complex adaptive systems. Within systems theory, complex adaptive systems act unpredictably on occasion, have tipping points that lead to fundamental reorganization, and can even display emergent properties that are difficult, if not impossible, to explain.
- Complex adaptive systems fail for a variety of reasons including incompetence or wrongdoing; design flaws and vulnerabilities; underestimating risks and failure to plan for low probability events; unforeseen high-impact events (Black Swans;[12] and what Charles Perrow characterized as uncontrollable and unavoidable "normal accidents" (discussed more fully here).
- Reasonable testing procedures will not be exhaustive and can fail to ascertain whether many complex adaptive systems will behave in an uncertain manner. Furthermore, the testing of complex systems is costly and only affordable by a few states, and they tend to be under pressure to cut military expenditures. To make matters worse, each software error fixed and each new feature added can alter a system's behavior in ways that can require additional rounds of extensive testing. No military can support the time and expense entailed in testing systems that are continually being upgraded.
- Learning systems can be even more problematic. Each new task or strategy learned can alter a system's behavior and performance. Furthermore, learning is not just a process of adding and altering information; it can alter the very algorithm that processes the information. Placing a system on the battlefield that can change its programming significantly raises the risk of uncertain behavior. Retesting dynamic systems that are constantly learning is impossible.
- For some complex adaptive systems various mathematical proofs or formal verification procedures have been used to ensure appropriate behaviors. Existing approaches to formal verification will not be adequate for

Calendar of Events

May 6–11
CHI'17: CHI Conference on Human Factors in Computing Systems,
Denver, CO,
Sponsored: ACM/SIG,
Contact: Susan R. Fussell,
Email: sfussell@cornell.edu

May 8–10
HotOS '17: Workshop on Hot Topics in Operating Systems,
Whistler, BC, Canada,
Sponsored: ACM/SIG,
Contact: Rachit Agarwal,
Email: ragarwal@cs.cornell.edu

May 10–12
GLSVLSI '17: Great Lakes Symposium on VLSI 2017,
Banff, AB, Canada,
Contact: Laleh Behjat,
Email: laleh@ucalgary.ca

May 15–17
CF'17: Computing Frontiers Conference,
Siena, Italy,
Sponsored: ACM/SIG,
Contact: Roberto Giorgi,
Email: giorgi@unisi.it

May 22–24
SYSTOR 2017:
International Systems and Storage Conference,
Haifa, Israel,
Sponsored: ACM/SIG,
Contact: Doron Chen,
Email: cdoron@il.ibm.com

May 24–26
SIGSIM-PADS '17:
SIGSIM Principles of Advanced Discrete Simulation
Singapore, Singapore,
Sponsored: ACM/SIG,
Contact: Wentong Cai,
Email: aswtcai@ntu.edu.sg

June

June 5–7
Web3D '17: The 22nd International Conference on Web3D Technology,
Brisbane, QLD, Australia,
Sponsored: ACM/SIG,
Contact: Matt Adcock,
Email: matt.adcock@csiro.au

June 5–8
ICMR '17: International Conference on Multimedia Retrieval,
Bucharest, Romania,
Sponsored: ACM/SIG,
Contact: Niculae Sebe,
Email: sebe@disi.unitn.it

systems with learning or planning capabilities functioning in complex socio-technical contexts. However, new formal verification procedures may be developed. The success of these will be an empirical question, but ultimately political leaders and military planners must judge whether such approaches are adequate for ensuring that LAWS will act within the constraints of IHL.

▸ While increasing autonomy, improving intelligence, and machine learning can boost the system's accuracy in performing certain tasks; they can also increase the unpredictability in how a system performs overall.

▸ Unpredictable behavior from a weapon system will not necessarily be lethal. But even a low-risk autonomous weapon will occasionally kill non-combatants, start a new conflict, or escalate hostilities.

Coordination, Normal Accidents, and Trust. Military planners often underestimate the risks and costs entailed in implementing weapon systems. Analyses often presume a high degree of reliability in the equipment deployed, and ease at integrating that equipment into a combat unit. Even autonomous weapons will function as components within a team that will include humans fulfilling a variety of roles, other mechanical or computational systems, and an adequate supply chain serving combat and non-combat needs.

Periodic failures or system accidents are inevitable for extremely complex systems. Charles Perrow labeled such failures "normal accidents."[8] The near meltdown of a nuclear reactor at Three Mile Island in Pennsylvania on March 28, 1979, is a classical example of a normal accident. Normal accidents will occur even when no one does anything wrong. Or they can occur in a joint cognitive system—where both operators and software are selecting courses of action—when it is impossible for the operators to know the appropriate action to take in response to an unanticipated event or action by a computational system. In the latter case, the operators do the wrong thing, because they misunderstand what the semi-intelligent system is trying to do. This was the case on December 6, 1999, when after a successful landing, confusion reigned, and a Global Hawk unmanned air vehicle veered off the runway and its nose collapsed in the adjacent desert, incurring $5.3 million in damages.[7]

In a joint cognitive system, when anything goes wrong, the humans are usually judged to be at fault. This is largely because of assumptions that the actions of the system are automated, while humans are presumed to be the adaptive players on the team. A commonly proposed solution to the failure of a joint cognitive system will be to build more autonomy into the computational system. This strategy, however, does not solve the problem. It becomes ever more challenging for a human operator to anticipate the actions of a smart system, as the system and the environments in which it operates become more complex. Expecting operators to understand how a sophisticated computer *thinks*, and to anticipate its actions so as to coordinate the activities of the team, increases the responsibility of the operators.

Difficulty anticipating the actions of other team members (human or computational) in turn undermines trust, an essential and often overlooked element of military preparedness. Heather Roff and David Danks have analyzed the challenges entailed for ensuring that human combatants will *trust* LAWS. For autonomous weaponry that have either planning capabilities or learning capabilities, they conclude that ensuring trust will require significant time, training, and cost.[10] This certainly does not rule out a satisfactory integration of LAWS into combat units. But it does suggest resources and costs that are seldom factored into determinations that autonomous systems are cost effective. Furthermore, there should be concerns as to whether appropriate training for combatants working with LAWS will actually be provided.

Since Perrow first proposed his theory of *normal accidents*, it has been fleshed out into a robust framework for understanding the safety of hazardous technologies. *Normal accident theory* is often contrasted to *high reliability theory*, which offers a more optimistic model for strategic planning.[11] Arguably good strategic planners would evaluate their proposed campaigns under the assumptions of both *high reliability theory* and *normal accident theory*. However, such comparisons can produce dramatically contrasting visions of the likelihood of success.

The unpredictability of complex adaptive systems, as partially captured in *normal accident theory*, underscores risks that might otherwise have been overlooked or ignored. This, however, is secondary to how much risk political leaders and military strategists consider acceptable.

Levels of Risk. As mentioned earlier, lethal autonomy is not a weapon system. It is a feature set that can be added to any weapon system. The riskiness of a specific LAW is largely a function of the destructiveness of the munition it carries.

Risk is commonly quantified as the probability of the event multiplied by its consequences. The risk posed by a weapon system rises relative to the power of the munitions the system can discharge, even when the likelihood of an adverse event occurring remains the same. Clearly, the immediate destructive impact of a machine gun pales in comparison to that of a nuclear warhead. The machine gun is inherently less risky.

Over time, increasingly sophisticated LAWS will be deployed. Therefore it behooves the member states of CCW to not be shortsighted in their evaluation of what will be a very broad class of military applications. CCW must not appear to green light autonomous systems that can detonate weapons of mass destruction. Given the high level of risks, powerful munitions, such as autonomous ballistic missiles or autonomous submarines capable of launching nuclear warheads, must be prohibited. Deploying systems that can alter their programming is also foolhardy. This last proviso would rule out many learning systems that, for example, improve their planning capabilities.

It may be difficult to accurately quantify whether a specific LAW is more or less reliable than a human.

States and military leaders may differ on the degree of unpredictability and level of risk they will accept in weapon systems. The risks posed by less powerful LAWS will, in all probability, be deemed acceptable to military strategists, particularly in comparison to similar risks posed by often unreliable humans. Nevertheless, it may be difficult to accurately quantify whether a specific LAW is more or less reliable than a human. While autonomous vehicles can be demonstrated to likely cause far fewer deaths than human drivers, similar benchmarks for accidents occurring during combat will be hard to collect and will be less than convincing. Perhaps, realistic simulated tests might demonstrate that LAWS outperformed humans in similar exercises. More importantly, the world has adjusted to accidents caused by humans. Public opinion is likely to be less forgiving of unintended wars or deaths of non-combatants caused by LAWS.

Regardless of the level of risk deemed acceptable, it is essential to recognize the degree of unpredictable risk actually posed by various autonomous weapons configurations. Empirical tools should be employed to adequately determine the risk posed by each type of LAW and whether that risk exceeds acceptable levels.

Most parties will agree that the unpredictability and therefore the risks posed by LAWS capable of dispatching high-powered munitions including nuclear weapons are unacceptable. The decision of states should not be whether any autonomous systems must be prohibited, but rather how broadly encompassing the prohibition on LAWS must be.

Mala in se

In the past, I have proposed that LAWS used for offensive purposes should be designated *mala in se*, a term coined by ancient Roman philosophers to designate an intrinsically evil activity. In just war theory and in IHL certain activities such as rape and the use of biological weapons are evil in and of themselves. Humanity's perception of evil can evolve. The ancient Romans did not consider slavery evil, but all civilized people do today. Machines that select targets and initiate lethal force are *mala in se* because they: "lack discrimination, empathy, and the capacity to make the proportional judgments necessary for weighing civilian casualties against achieving military objectives. Furthermore, delegating life and death decisions to machines is immoral because machines cannot be held responsible for their actions.[13]

Machines must not independently make choices or initiate actions that intentionally kill humans. Once this principle is in place, negotiators can move on to what will be a never-ending debate as to whether or when LAWS are extensions of human will and intention and under *meaningful human control*. With a strong moral principle in place it will be possible to condemn egregious acts.

The primary argument against this principle is the conjecture that future machines will display a capacity for discrimination and may even be more moral in their choices and actions than human soldiers.[1,2] Many in the AI and robotic community hope and believe that intelligent computational systems are becoming more than mere machines. That prospect, however, should not blind us to the opportunity to limit their destructive impact. If and when robots become ethical actors that can be held responsible for their actions, we can then begin debating whether they are no longer machines and are deserving of some form of legal personhood.

Conclusion

The short-term benefits of LAWS could be far outweighed by long-term consequences. For example, a robot arms race would not only lower the barrier to accidentally or intentionally start new wars, but could also result in a pace of combat that exceeds human response time and the reflective decision-making capabilities of commanders. Small low-cost drone swarms could turn battlefields into zones unfit for humans. The pace of warfare could escalate beyond meaningful human control. Military leaders and soldiers alike are rightfully concerned that military service will be expunged of any virtue.

In concert with the compelling legal and ethical considerations LAWS pose for IHL, unpredictability and risk concerns suggest the need for a broad prohibition. To be sure, even with a ban, bad actors will find LAWS relatively easy to assemble, camouflage, and deploy. The Great Powers, if they so desire, will find it easy to mask whether a weapon system has the capability of functioning autonomously.

The difficulties in effectively enforcing a ban are perhaps the greatest barrier to be overcome in persuading states that LAWS are unacceptable. People and states under threat perceive advanced weaponry as essential for their immediate survival. The stakes are high. No one wants to be at a disadvantage in combating a foe that violates a ban. And yet, violations of the ban against the use of biological and chemical weapons by regimes in Iraq and in Syria have not caused other states to adopt these weapons.

The power of a ban goes beyond whether it can be absolutely enforced. The development and use of biological and chemical weapons by Saddam Hussein helped justify the condemnation of the regime and the eventual invasion of Iraq. Chemical weapons use by Bashar al-Assad has been widely condemned, even if the geopolitics of the Syrian conflict have undermined effective follow-through in support of that condemnation.

The short-term benefits of LAWS could be far outweighed by long-term consequences.

A ban on LAWS is likely to be violated even more than that on biological and chemical weapons. Nevertheless, a ban makes it clear that such weapons are unacceptable and those using them are deserving of condemnation. Whenever possible that condemnation should be accompanied by political, economic, and even military measures that punish the offenders. More importantly, a ban will help slow, if not stop, an autonomous weapons arms race. But most importantly, banning LAWS will function as a moral signal that international humanitarian law (IHL) retains its normative force within the international community. Technological possibilities will not and should not succeed in pressuring the international community to sacrifice, or even compromise, the standards set by IHL.

A ban will serve to inhibit the unrestrained commercial development and sale of LAWS technology. But a preemptive ban on LAWS will not stop nor necessarily slow the roboticization of warfare. Arms manufacturers will still be able to integrate ever-advancing features into the robotic weaponry they develop. At best, it will require that a human in the loop provides a real-time authorization before a weapon system kills or destroys a target that may harbor soldiers and noncombatants alike.

Even a modest ban signals a moral victory, and will help ensure that the development of AI is pursued in a truly beneficial, robust, safe, and controllable manner. Requiring meaningful human control in the form of real-time human authorization to kill will help slow the pace of combat, but will not stop the desire for increasingly sophisticated weaponry that could potentially be used autonomously.

In spite of recent analyses suggesting that humanity has become less violent over several millennia,[9] warfare itself is an evil humanity has been unsuccessful at quelling. However, if we are to survive and evolve as a species some limits must be set on the ever more destructive and escalating weaponry technology affords. The nuclear arms race has already made clear the dangers inherent in surrendering to the inevitability of technological possibility.

Arms control will never be a simple matter. Nevertheless, we must slowly, effectively, and deliberately put a cap on inhumane weaponry and methods as we struggle to transcend the scourge of warfare.

References

1. Arkin, R. The case for banning killer robots: Counterpoint. *Commun. ACM 58*, 12 (Dec. 2015), 46–47.
2. Arkin, R. *Governing Lethal Behavior in Autonomous Systems.* CRC Press, Boca Raton, FL, 2009.
3. Asaro, P. On Banning Autonomous Lethal Systems: Human Rights, Automation and the Dehumanizing of Lethal Decision-making. Special Issue on New Technologies and Warfare. *International Review of the Red Cross 94*, 886 (Summer 2012), 687–709.
4. Carpenter, C. How do Americans feel about fully autonomous weapons? The Duck of Minerva (June 19, 2013); http://bit.ly/2mBKMnR
5. Gormley, D.M. *Missile Contagion.* Praeger Security International, 2008.
6. Gubrud, M. and Altmann, J. Compliance Measures for an Autonomous Weapons Convention, ICRAC Working Paper Series #2, International Committee for Robot Arms Control (2013); http://bit.ly/2nfOLFu
7. Peck, M. Global hawk crashes: Who's to blame? *National Defense 87*, 594 (2003); http://bit.ly/2mQJgeJ
8. Perrow, C. *Normal Accidents: Living With High-Risk Technologies.* Basic Books, New York, 1984.
9. Pinker, S. *The Better Angels of Our Nature: Why Violence Has Declined.* Penguin, 2011.
10. Roff, H. and Danks, D. Trust but Verify: The difficulty of trusting autonomous weapons systems. *Journal of Military Ethics.* (Forthcoming).
11. Sagan, S.D. *The Limits of Safety: Organizations, Accidents, and Nuclear Weapons.* Princeton University Press, Princeton, NJ, 2013.
12. Taleb, N.N. *The Black Swan: The Impact of the Highly Improbable.* Random House, 2007.
13. Wallach, W. Terminating the Terminator. *Science Progress*, 2013; http://bit.ly/2mjl2dy
14. Wallach, W. and Allen, C. Framing robot arms control. *Ethics and Information Technology 15*, 2 (2013), 125–135.

Wendell Wallach (wendell.wallach@yale.edu) is a Senior Advisor to The Hastings Center and Chairs Technology and Ethics Studies at the Yale University Interdisciplinary Center for Bioethics. His latest book is *A Dangerous Master: How to Keep Technology from Slipping Beyond Our Control.*

The author would like to express appreciation to the five anonymous reviewers, whose comments contributed significantly to help clarify the arguments made in this Viewpoint.

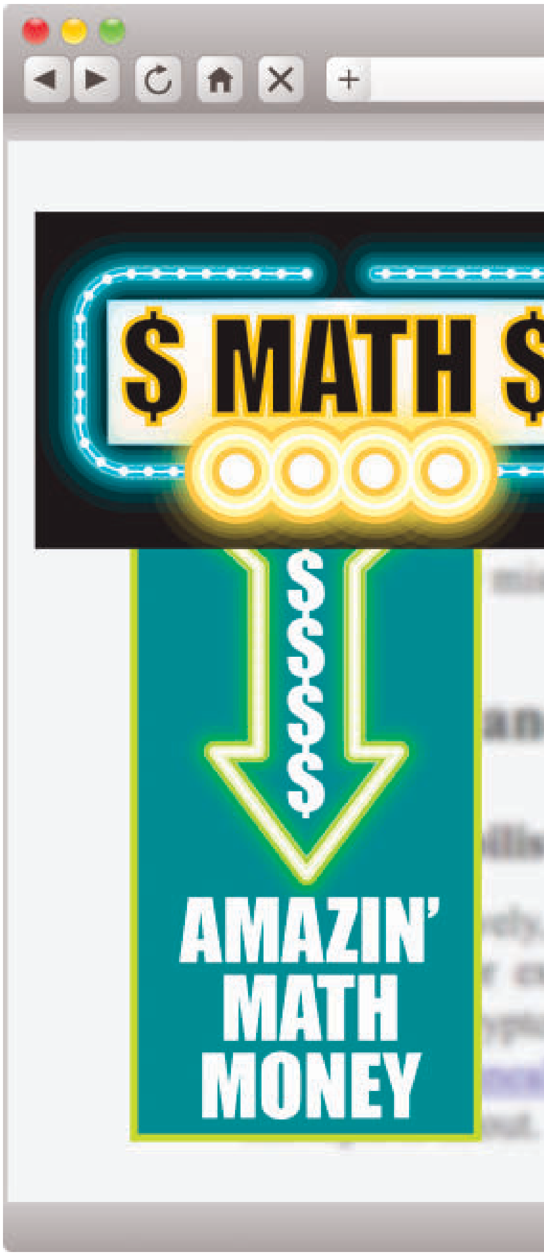
DOI:10.1145/3052935

Article development led by acmqueue
queue.acm.org

Modern applications are increasingly using probabilistic machine-learned models.

BY ERIK MEIJER

Making Money Using Math



If Google were created from scratch today, much of it would be learned, not coded.

—Jeff Dean, Google Senior Fellow, Systems and Infrastructure Group

MACHINE LEARNING, OR ML, is all the rage today, and there are good reasons for that. Models created by machine-learning algorithms for problems such as spam filtering, speech and image recognition, language translation, and text understanding have many advantages over code written by human developers. Machine learning, however, is not as magical as it sounds at first. In fact, it is rather analogous to how human developers create code using test-driven development.[4] Given a training set of input-output pairs $\{(a,b) | a \in A, b \in B\}$, guess a function $f \in A \to B$ that passes all the given tests but also generalizes to unspecified input values.

A big difference between human-written code and learned models is that the latter are usually not represented by text and hence are not understandable by human developers or manipulable by existing tools. The consequence is that none of the traditional software engineering techniques for conventional programs such as code reviews, source control, and debugging are applicable anymore. Since incomprehensibility is not unique to learned code, these aspects are not of concern here.

A more interesting divergence between machines and humans is that machines are less arrogant than humans, and they acknowledge uncertainty in their code by returning a *probability distribution* or *confidence interval* of possible answers $f \in A \to \mathbb{P}(B)$ instead of claiming to know the precise result for every input. For example, a learned image-recognition function

COLLAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

by a major cloud provider will predict with 95% certainty that I have hair, but is less confident about *whether or not* I am professional (Figure 1).

The implication of incorporating learned models in human-written code is that you cannot get around the fact that the building blocks from which humans compose applications are fundamentally probabilistic. This is a challenge for mainstream programming languages, which all assume that computations are precise and deterministic. Fortunately, the 18th-century Presbyterian minister Thomas Bayes anticipated the need for dealing with uncertainty and formulated Bayes' rule:[6]

$$\mathbb{P}(\mathtt{A}|\mathtt{B})*\mathbb{P}(\mathtt{B}) = \mathbb{P}(\mathtt{A}\&\mathtt{B}) = \mathbb{P}(\mathtt{B}|\mathtt{A})*\mathbb{P}(\mathtt{A})$$

As it turns out, Bayes' rule is exactly what the doctor ordered when it comes to bridging the gap between ML and contemporary programming languages.

Many of the mathematical explanations of Bayes' rule are deeply confusing for the working computer scientist, but, remarkably, when interpreted from a functional programming point of view, *Bayes' rule is a theorem about composability and invertibility of monadic functions*. Let's break down Bayes' rule piece by piece and rebuild it slowly based on developer intuition.

Probability Distributions

First let's explore what probability distributions $\mathbb{P}(\mathtt{A})$ are. The Wikipedia definition, "*a probability distribution is a mathematical description of a random phenomenon in terms of the probabilities of events*," is rather confusing from a developer perspective. If you click around for a bit, however, it turns out that a discrete distribution is just a generic list of pairs of values and probabilities $\mathbb{P}(\mathtt{A})=[\mathtt{A} \mapsto \mathbb{R}]$ such that the probabilities add up to 1. This is the *Bayesian representation* for distributions. Isomorphically, you can use the *frequentist representation* of distributions as infinite lists of type `dist` ∈ `[A]`, as n gets larger, sampling from the collection and counting the frequencies of each element `from a in dist.Take(n) group by a into g select g.Key a g.Sum()/n` approximates the Bayesian representation of the distribution. When converting from the Bayesian to the frequentist implementation, the probabilities do not to have to add up to 1, and the sampling process will ensure the ratios are properly normalized.

Like true mathematicians, we will silently switch between these two representations of distributions whenever convenient. Unlike mathematicians, however, to keep things simple we will not consider continuous distributions. We want our distribution to hold generically any type `A`, and most of the types we deal with in code are discrete and not "measurable" or real number-like.

Because the values we care about are usually not even comparable, we also will avoid cumulative distributions. One reason that mathematicians like standard continuous distributions—such as Gaussian, beta, binomial, and uniform—is because of their nice algebraic properties, called *conjugate priors*.[2] For example, uniform prior combined with a binomial likelihood results in a beta posterior. This makes 18th- and 19th-century probabilistic computations using pencil and paper feasible, but that is not necessary now that there are powerful computers that can run millions of simulations per second.

In programming examples, distributions typically come from outside data as discrete frequentist collections of data with an unknown distribution, or they are defined explicitly as a Bayesian representation by enumerating a list of value/probability pairs. For example, here is the distribution of weight of adults in the United States, according to the Centers for Disease Control (CDC):

```
CDC ∈ ℙ (Weight)
CDC = [obese ↦ 0.4, skinny ↦ 0.6]
```

Efficiently sampling from composed distributions is, indeed, rocket science. Just like database query optimizers, advanced sampling methods leverage properties of the leaf distributions and the structure of the query[9] or program[3] that computes the distribution. It leverages deep and complex mathematical techniques such as importance sampling, Metropolis-Hastings, Markov Chain Monte Carlo, and Gibbs sampling that are far outside the scope of this article but are important for making real-world computations over probability distributions feasible. As Bayesian analysis consultant John D. Cook remarked "... Bayesian statistics goes back over 200 years, but it did not take off until the 1980s because that's when people discovered practical numerical methods for computing with it ..."

To illustrate the sophistication involved in efficiently sampling known discrete distributions, imagine converting the example distribution CDC

Figure 1. Image recognition results.

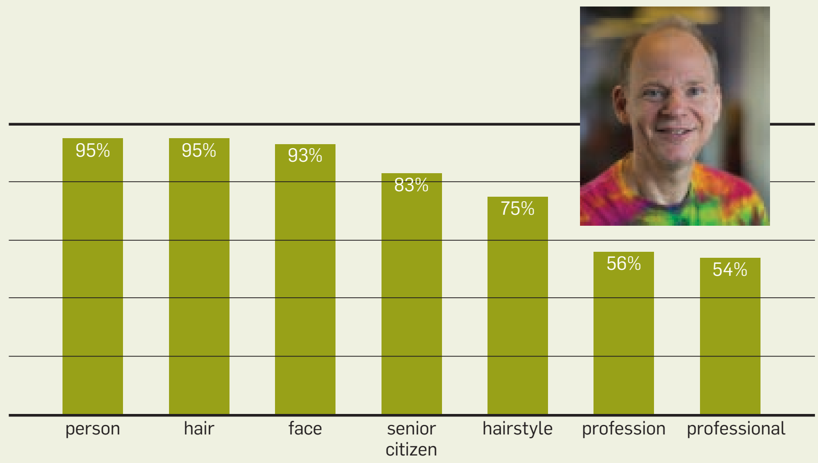


Figure 2. Frequentist representation.

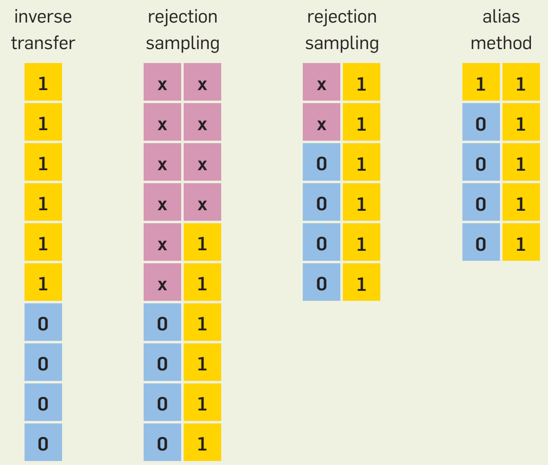


Figure 3. Joint probability distribution.

P(food & weight)	burger	celery	P(weight)
obese	0.4*0.9 = 0.36	0.4*0.1 = 0.04	0.36+0.04 = 0.4
skinny	0.6*0.3 = 0.18	0.6*0.7 = 0.42	0.18+0.42 = 0.6
P(food)	0.36+0.18 = 0.54	0.04+0.42 = 0.46	

into a frequentist representation.[8]

Perhaps the most obvious method stacks the columns for skinny and obese on top of each other and draws one random number—say, p—between 0 and 1 and then checks if p ≤ 0.4 yields obese, and otherwise yields skinny. In general, this search is linear in the number values in the distribution, but using tricks like binary search tree can speed things up. Mathematicians call this the *inverse transfer* method.

Another way is first to select a random integer—say, weight—to select between obese and skinny, and then choose a random double between 0 and 1—say, *p*—and if CDC[weight] ≤ p, then yield weight, as shown in Figure 2. Mathematicians call this algorithm rejection sampling, and as the histogram shows, half of the attempts to sample a value from the distribution will fail (the pink part). This can be improved by picking a tighter envelope distribution, like that in the second histogram, but that still rejects two out of 12 samples.

The last method pads the lower probabilities by borrowing from the higher probabilities. Amazingly, it is always possible to do this in a way such that every column represents the probabilities for, at most, two values, so we need only one comparison to pick the right value. This comparison can be implemented using a second index table, and hence mathematicians call this sampling algorithm the *alias method*.

Efficiently sampling from composed distributions is, indeed, rocket science.

Conditional Probability Distributions

Now that we have explained probability distributions ℙ(A), let's examine conditional probability distributions ℙ(B|A), which, according to Wikipedia, are "a measure of the probability of an event given that (by assumption, presumption, assertion, or evidence) another event has occurred." To developer ears that sounds exactly like a function A→ℙ(B) that returns a distribution, just like a learned model. The remainder of this article uses the notations ℙ(B|A) and A→P(B) interchangeably.

Going back to the example, we have the following model Doctor ∈ ℙ(Food|Weight) of food preference, given weight, that we could have obtained by asking patients what kind of food they consume:

```
Doctor ∈ℙ(Food|Weight)
   = Weight → ℙ(Food)
Doctor(obese)
   = [burger↦0.9, celery↦0.1]
Doctor(skinny)
   = [burger↦0.3 celery↦0.7]
```

As argued in the introduction, these probabilistic functions, such as ℙ(Object|Image), ℙ(Text|Audio), ℙ(Spam|Text), and so on, increasingly are the result of training some ML algorithm or neural net, instead of being coded by expensive and flaky human developers.

Now that you know that conditional probabilities are probabilistic functions, things are starting to get interesting, since this means that multiplication (*) used in Bayes' rule is an operator that applies a probabilistic function to a probability distribution as a parameter—that is, it has the following type signature:

```
ℙ(B|A)*ℙ(A) ∈ℙ(A&B)
```

Using the Bayesian representation of distributions, you can implement a probabilistic function application likelihood*prior where likelihood∈ℙ(B|A) and prior∈ℙ(A), using the following correlated query:

```
likelihood*prior =
    from a↦p in prior
    from b↦q in likelihood(a)
    select (a,b)↦p*q
```

Applying this definition to compute the result of Doctor*CDC, we obtain the table shown in Figure 3 for the joint probability distribution ℙ(Food&Weight).

Because the distributions for ℙ(Weight) and ℙ(Food) appear in the margins of this table, mathematicians call them *marginal probabilities*, and similarly the process of summing up the columns/rows is called *marginalization*. When computing a joint distribution using (*), mathematicians often use the name *likelihood* for the function and *prior* for the argument.

The beauty of the frequentist representation is that there is no need for multiplying probabilities. Sampling ensures the underlying ratio of occur-

Figure 4. Syntactic sugar.

from a in prior	from a→p in prior
from b in likelihood (a)	from b→q in likelihood (a)
select a⊕b	select a⊕b→p*q

rence of values in the result will automatically reflect the proper product of values from the prior and likelihood. For example, if we implement the prior CDC by an infinite collection with odds obese:skinny = 4:6, and the result of Doctor(skinny) by an infinite collection with odds burger:celery = 3:7, and, respectively, that of Doctor(obese) by a collection with odds burger:celery = 9:1, then sampling from the infinite collection Doctor*CDC, which results from applying the prior to the likelihood, will have a ratio:

```
(obese:burger):
(obese,celery):(skinny,burger):
(skinnny:celery) = 36:4:18:24.
```

The keen reader will note that (*) is a slight variation of the well-known monadic bind operator, which, depending on your favorite programming language, is known under the names (>>=), SelectMany, or flatMap. Indeed, *probability distributions form a monad*. Mathematicians call it the Giry monad, but Reverend Bayes beat everyone to it by nearly two centuries.

Note that as formulated, Bayes' rule has a type error that went unnoticed for centuries. The left-hand side returns a distribution of pairs ℙ(A&B), while the right-hand side returns a distribution of pairs ℙ(B&A). Not a big deal for mathematicians since & is commutative. For brevity we'll be sloppy about this as well. Since we often want to convert from ℙ(A&B) to ℙ(A) or ℙ(B) by dropping one side of the pair, we prefer the C#-variant of SelectMany that takes a combiner function A⊕B∈C to post-process the pair of samples from the prior and likelihood:

```
likelihood*prior =
    from a↦p in prior
    from b↦q in likelihood(a)
    select a⊕b↦p*q
```

Now that we know that (*) is monadic bind, we can start using syntactic sugar such as LINQ queries or for/monad comprehensions. All that is really saying is that it is safe to drop the explicit tracking of probabilities from any query written over distributions (that is, the code on the left in Figure 4 is simply sugar for the code on the right, which itself can be alternatively implemented with the frequentist approach using sampling).

Another way of saying this is that we can use query comprehensions as a DSL (domain-specific language) for specifying probabilistic functions. This opens the road to explore other standard query operators besides application that can work over distributions and that can be added to our repertoire. The first one that comes to mind is filtering, or *projection* as the mathematicians prefer to call it.

Given a predicate (A→𝔹), we can drop all values in a distribution for which the predicate does not hold using the division operator (÷):

```
ℙ (A)÷(A→𝔹) ∈ ℙ(A)
prior÷condition = from a in prior
where condition(a) select a
```

The traditional division of a distribution ℙ(A&B) by distribution ℙ(B) can be defined similarly as

```
joint ÷ evidence =
    λb.from (a,b) in joint from
    b' in evidence where b=b'
    select (a,b)
```

We can show that (f*d)÷d = f. Applying the latter version to Bayes' rule results in the following equivalence:

```
ℙ(A|B) = ℙ(B|A)* ℙ(A) ÷ ℙ(B)
```

In practice, it is most convenient to use query comprehensions directly instead of operators, and write code like this:

```
Posterior ∈ ℙ(C|B)=B→ ℙ(C)
posterior(b) =
   from a in prior
   from b' in likelihood(a) where
   b = b'
   select a⊕b
```

Whichever way you spin it, this is incredibly cool! Bayes' rule shows how to invert a probabilistic function of type ℙ(B|A) = A→ℙ(B)into a probabilistic function of type ℙ(A|B) = B→ℙ(A) using conditioning.

When function inversion is applied to the running example, probabilistic function PredictWeightFromFood ∈ ℙ(Weight|Food) can be defined as follows:

```
PredictWeightFromFood
   ∈ Food→ℙ(Weight)
PredictWeightFromFood(food)
   =(Doctor*CDC) ÷ ( _ = food)
```

Removing all syntactic sugar and using the value/probability pairs implementation amounts to the following probabilistic function:

```
PredictWeightFromFood
  ∈ Food→ℙ(Weight)
PredictWeightFromFood(burger)
  = [obese↦36, skinny↦18]
PredictWeightFromFood(celery)
  = [obese↦4, skinny↦42]
```

In practice, most monads have an unsafe run function of type ℙ(A)→M(A) that teleports you out of the monad into some concrete container M. Mathematicians call this the *forgetful functor*. For distributions dist ∈ ℙ(A), a common way to exit the monad is by picking the value a ∈ A with the highest probability in dist. Mathematicians use the higher-order function arg max for this, and call it MLE (maximum likelihood estimator) or MAP (maximum a posteriori). In practice it is often more convenient to return the pair a↦p from dist with the highest probability.

A simple way to find the value with the maximal likelihood from a frequentist representation of a distribution is to blow up the source distribution ℙ(A) into a distribution of distributions ℙ(ℙ(A)), where the outer distribution is an infinite frequentist list of inner Bayesian distributions [A↦ℝ], computed by grouping and summing, that over time will converge to true underlying distribution. Then you can select the *n*th inner distribution and take its maximum value.

```
WeightFromFood ∈ Food → [A↦ℝ]
WeightFromFood food
    =PredictWeightFromFood(food).
    Run().ElementAt(1000)
```

Using the query-style DSL for composing and conditioning probabilistic functions is great, but it falls short of being a real programming language with arbitrary control flow, loops, try/catch blocks, recursion, among others. Since distributions are a variant of the continuation monad, it is possible to integrate probabilistic computations into a regular imperative language similar to the async await syntax now available in many programming languages. An example of an imperative *probabilistic programming language* is WebPPL (http://webppl.org), which embeds the distribution monad into regular JavaScript. In WebPPL, the running example looks as follows:

```
var cdc = function() {
 return Categorical({ ps: [4, 6],
   vs: [“obese”, “skinny”] })
}
var doctor = function(weight) {
  if(“obese” == weight)
   return Categorical({ ps: [9, 1],
     vs: [“burger”, “celery”] } })
  if(“skinny” == weight)
   return Categorical({ ps: [3, 7],
     vs: [“burger”, “celery”] } })
}
var predict = function(food) {
  var weight = sample(cdc())
  var food _ =
sample(doctor(weight))
  condition(food == food _ )
  return weight;
}
```

The assignment + sample statement

```
var a = sample(prior)
… rest of the program…
```

is exactly like the query fragment

```
from a in prior
… rest of the query …
```

and randomly picks a value a ∈ A from a distribution prior ∈ ℙ(A). The condition(p) statement corresponds to a where clause in a query.

To “run” this program, we pass the predict function into the WebPPL inference engine as follows:

```
Infer({method: enumerate,
samples: 10000},
function(){return
predict(“burger”)})
```

This samples from the distribution described by the program using the Infer function with the specified sampling method (which includes enumerate, rejection, and MCMC) that reifies the resulting distribution into a Bayesian representation.

Applications of Probabilistic Programming

Suppose ordinary developers had access to a probabilistic programming language. What scenarios would this open up?

If we take a step back and look at a typical Web or mobile application, it implements the standard reinforcement learning design pattern shown in Figure 5. We have to predict an action to send to the user, based on the user's state and the dollar value extracted from the user, such that the sum of the rewards over time is maximized.

For games such as AlphaGo,[10] the agent code is often a neural network, but if we abstract the pattern to apply to applications as a whole, it is likely a combination of ML learned models and regular imperative code. This hybrid situation is true even today where things such as ad placement and search-result ranking are probabilistic but opaquely embedded into imperative code. Probabilistic programming and machine learning will allow developers to create applications that are highly specialized for each user.

One of the attractions of IDEs (integrated development environments) is autocomplete, where the IDE predicts what a user is going to type, based on what has been typed thus far. In most IDEs, autocomplete is driven by static type information. For example, if the user types ppl, the JetBrains Rider IDE shows all the properties of the string type as potential completions, as shown in Figure 6.

Note that the completion list is shown in deterministic alphabetical order, rather than being probabilistically ranked using some learned model based on which methods on string are the most likely in the given context. Hence, the IDE should implement autocomplete using a probabilistic function autoComplete ∈ ℙ([Completion]|Context) that returns a distribution of possible completions based on the current user context.[7] Another recent application of ML and probabilistic programming in the compiler space is to infer pretty-print rules by learning from patterns in a large corpus of code prettyPrint ∈ ℙ(String|AST).[5]

For an example application of exposing the representation of distributions, let's revisit the feedback loop between user and application.

Figure 5. Standard reinforcement learning design pattern.

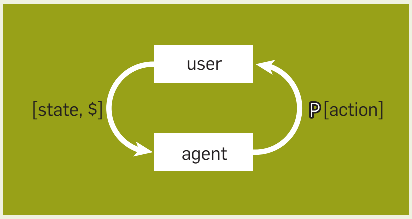
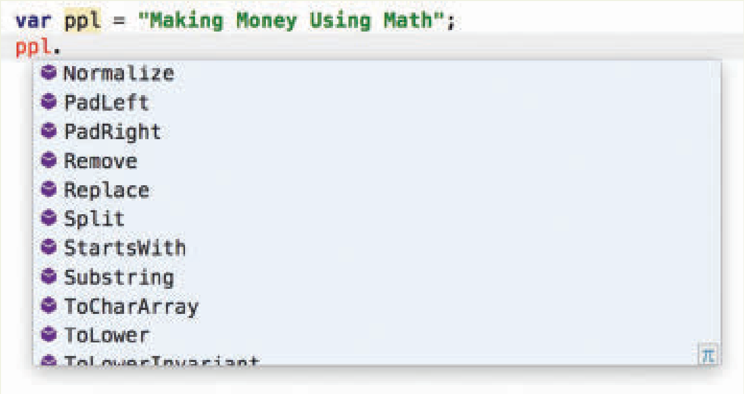


Figure 6. Autocomplete example.



```
var ppl = "Making Money Using Math";
ppl.
  Normalize
  PadLeft
  PadRight
  Remove
  Replace
  Split
  StartsWith
  Substring
  ToCharArray
  ToLower
  ToLowerInvariant
```

Determine the optimal title for this article that would maximize click-through on the *CACM* website. That is, should we use "Probabilistic Programming for Dummies" instead of the current title "Making Money Using Math?"

In this case, we create the model shown in Figure 7, the set of all users as a conditional distribution of a user clicking on the article given the title:

$$\texttt{User} \in \mathbb{P}(\texttt{Click}|\texttt{Title})$$

Note we do not want to make any a priori assumptions about the underlying distributions other than the frequentist stream of clicks received, given the frequentist stream of titles served to the users.

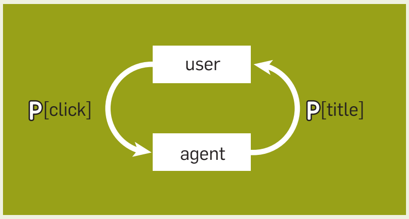
The agent in this case wants to find out over time which possible title for a story will generate the most clicks from the users, and hence we will model the agent by the higher-order function that takes the users and from that creates a distribution of titles:

$$\texttt{agent} \in (\texttt{Title}\to\mathbb{P}(\texttt{Click}))\to\mathbb{P}(\texttt{Title})$$

Mathematicians call the implementation of `user` a *Bayesian bandit*,[11] and they leverage the fact that Bernoulli and beta distributions are conjugate priors.[12] They call the variant of the `run` function we will be using *Thompson sampling*.[1]

When viewed from a computer scientist's point of view, the operational solution is relatively straightforward. We convert the user behavior that returns a distribution of clicks $\texttt{user} \in \texttt{Title}\to\mathbb{P}(\texttt{Click})$ into a function $\texttt{Title}\to\mathbb{P}(\texttt{Title\&Click}\mapsto\mathbb{R})$ that returns a distribution of pairs of titles and clicks using `run` as explained earlier (this corresponds to the beta distribution part of the algorithm. We do not track the "uncertainty" about $\mathbb{P}(\texttt{Click})$, but we can easily compute that together with the click probability if that is useful). A small tweak is needed in that we are interested only in clicks that are true, and not in those that are `false` (this is the Bernoulli part of the algorithm).

Figure 7. Set of all users as a conditional distribution.



This allows us to observe how the probability that the user will click on each title evolves over time as we see more clicks from the users. Whenever we need to produce a new title, we use the `Title` for which the most recent $\texttt{Title\&Click}\mapsto\mathbb{R}$ has the highest probability (this is the Thompson sampling part of the algorithm). In other words, the Bayesian bandit is essentially a merge sort over the reified underlying probability distributions of the clicks from the users.

The computational model underneath modern applications such as self-driving cars, speech and image recognition, personalized recommendations, and so on, is changing from classical deterministic computations to probabilistic machine-learned models. Currently, building such applications requires deep knowledge and expertise of the underlying details of the ML algorithms using custom tools.

Conclusion

Probabilistic programming aims to democratize the application of machine learning by allowing regular programmers to incorporate ML in general-purpose programming languages without friction. As illustrated in this article, from a semantics point of view, a probabilistic language simpl y adds the probability monad to the set of ambient side effects and leverages Bayes' rule to compose and condition probability distributions. Efficiently implementing probabilistic languages and providing the proper software engineering tooling, however, will keep compiler and programming-language experts busy for a long time. C

Related articles on queue.acm.org

Information Extraction: Distilling Structured Data from Unstructured Text
Andrew McCallum
http://queue.acm.org/detail.cfm?id=1105679

Accountability in Algorithmic Decision-making
Nicholas Diakopoulos
http://queue.acm.org/detail.cfm?id=2886105

Learning from THE WEB
Adam Bosworth
http://queue.acm.org/detail.cfm?id=1103833

References

1. Agrawal, S., Goyal, N. Analysis of Thompson sampling for the multi-armed bandit problem. *J. Machine Learning Research: Workshop and Conference Proceedings 23* (2012); http://jmlr.org/proceedings/papers/v23/agrawal12/agrawal12.pdf.
2. Fink, D. A compendium of conjugate priors. Montana State University, 1997; http://www.johndcook.com/CompendiumOfConjugatePriors.pdf.
3. Goodman, N.D. The principles and practice of probabilistic programming. In *Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2013; http://dl.acm.org/citation.cfm?id=2429117.
4. Norvig, P. Machine learning for programming. *InfoQueue*, 2015; https://www.infoq.com/presentations/machine-learning-general-programming.
5. Parr, T., Vinju, J. Towards a universal code formatter through machine learning. In *Proceedings of the ACM SIGPLAN International Conference on Software Language Engineering*, 2016; http://dl.acm.org/citation.cfm?id=2997383.
6. Paulos, J.A. The mathematics of changing your mind. *New York Times Sunday Book Review*; http://www.nytimes.com/2011/08/07/books/review/the-theory-that-would-not-die-by-sharon-bertsch-mcgrayne-book-review.html?_r=0.
7. Raychev, V., Bielik, P., Vechev, M. Probabilistic model for code with decision trees. In *Proceedings of the ACM SIGPLAN International Conference on Object-oriented Programming, Systems, Languages, and Applications*, 2016; http://dl.acm.org/citation.cfm?doid=2983990.2984041.
8. Schwarz, K. Darts, dice, and coins: Sampling from a discrete distribution, 2011; http://www.keithschwarz.com/darts-dice-coins/.
9. Scibior, A., Ghahramani, A., Gordon, A.D. Practical probabilistic programming with monads. In *Proceedings of the 2015 ACM SIGPLAN Symposium on Haskell*; http://dl.acm.org/citation.cfm?id=2804317.
10. Silver, D., et al. Mastering the game of Go with deep neural networks and tree search, 2016; https://blog.acolyer.org/2016/09/20/mastering-the-game-of-go-with-deep-neural-networks-and-tree-search/.
11. Stucchio, C. Bayesian bandits—Modifying click-throughs with statistics, 2013; https://www.chrisstucchio.com/blog/2013/bayesian_bandit.html.
12. Wikipedia. Conjugate prior: Discrete distributions; https://en.wikipedia.org/wiki/Conjugate_prior#Discrete_distributions.

Erik Meijer has been working on "Democratizing the Cloud" for the past 15 years. He is known for his work on the Haskell, C#, Visual Basic, and Dart programming languages, among others, as well as for his contributions to LINQ and the Reactive Framework (Rx).

DOI:10.1145/3052937

Article development led by acmqueue
queue.acm.org

The fuzzer is for those edge cases that your testing did not catch.

BY ROBERT GUO

MongoDB's JavaScript Fuzzer

AS MONGODB BECOMES more feature-rich and complex with time, the need to develop more sophisticated methods for finding bugs grows as well. Three years ago, MongDB added a home-grown JavaScript fuzzer to its toolkit, and it is now our most prolific bug-finding tool, responsible for detecting almost 200 bugs over the course of two release cycles. These bugs span a range of MongoDB components from sharding to the storage engine, with symptoms ranging from deadlocks to data inconsistency. The fuzzer runs as part of the continuous integration (CI) system, where it frequently catches bugs in newly committed code.

Fuzzing, or fuzz testing, is a technique for generating *randomized, unexpected, and invalid input* to a program to trigger untested code paths. Fuzzing was originally developed in the 1980s and has since proven to be effective at ensuring the stability of a wide range of systems, from file systems[15] to distributed clusters[10] to browsers.[16] As people have attempted to make fuzzing more effective, two philosophies have emerged: smart and dumb fuzzing. As the state of the art evolves, the techniques that are used to implement fuzzers are being partitioned into categories, chief among them being *generational* and *mutational*.[7] In many popular fuzzing tools, smart fuzzing corresponds to generational techniques, and dumb fuzzing to mutational techniques, but this is not an intrinsic relationship. Indeed, in our case at MongoDB, the situation is precisely reversed.

Smart Fuzzing

A smart fuzzer is one that has a good understanding of the valid input surface of the program being tested. With this understanding, a smart fuzzer can avoid getting hung up on input validation and focus on testing a program's behavior. Testing that a program properly validates its input is important but isn't the goal of fuzz testing.

Many fuzzers rely on an explicit grammar to generate tests, and it is that grammar that makes those tests smart. But MongoDB's command language

Figure 1. AST of one command in the corpus.

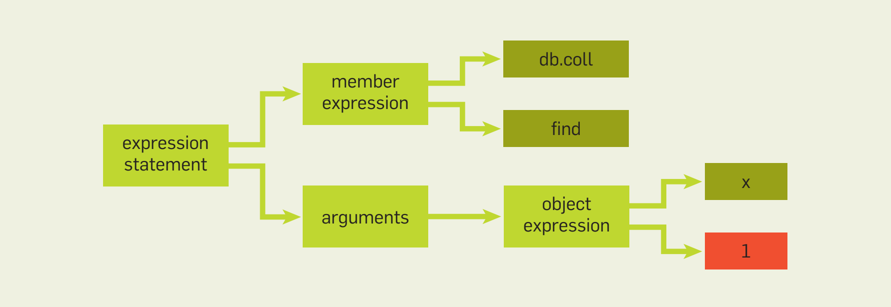


Figure 2. Node replaced with placeholder node.

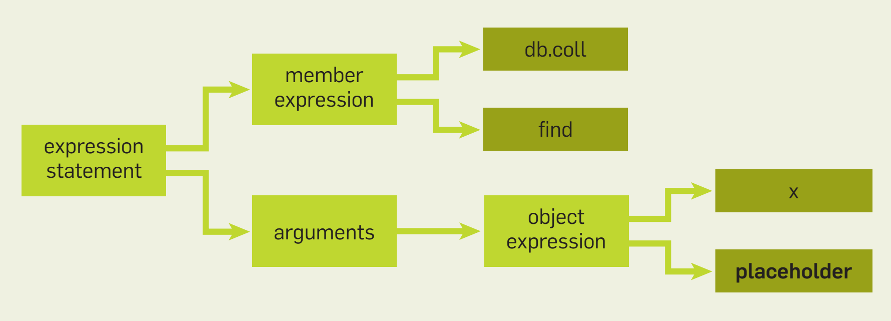
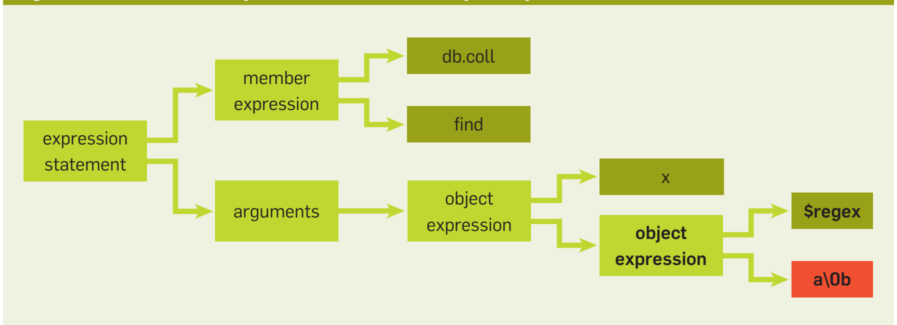


Figure 3. Placeholder replaced with another ObjectExpression.



is young, and we did not want to delay our fuzzer's delivery by taking the time to distill a formal grammar. Instead, we borrow our knowledge of the MongoDB command grammar from our corpus of existing JavaScript integration tests,[18] mutating them randomly to create novel test cases. Thus, our *mutational* strategy results in a *smart* fuzzer.

These JavaScript integration tests have been a mainstay of our testing for many years. Our CI system, Evergreen,[8] invokes a test runner that feeds each test file to a mongo shell, which executes the commands within the test file against MongoDB servers, shard routers, and other components to be tested. When the fuzzer runs, it takes in a random subset of these JS tests and converts them to an AST (abstract syntax tree) of the form understood by JavaScript interpreters. It then wreaks (controlled) havoc on the tree by selectively replacing nodes, shuffling them around, and replacing their values. This way we generate commands with parameters that wouldn't be encountered during normal testing but preserve the overall structure of valid JavaScript objects.

For example, the code `db.coll.find({x:1})` finds a document in collection `coll` with a field `x` having the value `1`, as shown in Figure 1.

To begin fuzzing that AST, the fuzzer first traverses the tree to mark nodes that should be replaced. In this case, assume it has decided to replace the value of the `ObjectExpression`,[14] a `1`. This node is then replaced with a placeholder node, as shown in Figure 2.

As the fuzzer traverses the tree, it also picks up values that it thinks are interesting, which are usually primitive values such as strings and numbers. These values are harvested and used to construct the final values of the placeholder nodes.

In this example, the placeholder is replaced with another `ObjectExpression` containing the key and value that it harvested from elsewhere in the corpus, as shown in Figure 3.

When this tree is converted into code, it becomes a new test case:

```
db.coll.find(x:{$regex:’ab’}}
```

This replaces the original test case of finding a document whose field `x` has value `1` with a new test case that finds documents matching the regular expression `a\0b`.

A test very much like this one was actually run by our fuzzer, and it turned out that MongoDB did not properly handle regular expressions strings containing null bytes, so this test case caused the server to crash.[11]

Lessons From Experience

AST > Regular expressions

Using an abstract syntax tree is a great strategy for fuzz testing. Previously, we had tried fuzzing using a regex-based approach. This involved stringifying the tests and finding specific tokens to replace or shuffle. Maintaining those regexes became a nightmare after a while, and it's very easy to introduce subtle mistakes that cause the mutations to become less effective. Syntax trees, on the other hand, are designed to represent all the information you need to know about the code, which is a superset of what can be deduced from using regexes. Additionally, ASTs are very difficult to get wrong: all the fuzzer is doing is manipulating properties in an object.

Open source libraries that turn code into ASTs for most languages are available; we used acorn.js.[1]

Heuristic > Random

When implementing the mutational aspect of a fuzzer, noting which types of mutations are provoking the most bugs can yield benefits. The initial implementation randomly chose which nodes to replace, but modified `ObjectExpressions` contributed to finding more new bugs, so we tweaked the probabilities to make more mutations happen on ObjectExpressions.

Dumb Fuzzing

Smart, AST-based mutation gives the MongoDB fuzzer a familiarity with the input format, but it also guarantees blind spots, because the corpus is a finite list harvested from human-written tests. The school of dumb fuzzing proposes an answer to this shortcoming, advocating fuzzers that generate input randomly, without regard to validity, thereby covering areas the developer may have overlooked.

This is a bit of a balancing act. With no knowledge of the target program at all, the best a fuzzer could do would be to feed in a random stream of 0s and 1s. That would generally do nothing but trigger input validation code at some intervening layer before reaching the program under test. Triggering only input validation code is the hallmark of a bad fuzzer.

To put some dumb in our fuzzer without resorting to random binary, values are generated from a seed list. Since our test inputs are JavaScript objects consisting of MongoDB commands and primitive values, the seed list is composed of MongoDB commands and primitive types that we know from experience are edge cases. These seed values are kept in a file, and JavaScript objects are generated using them as the keys and values. Here's an excerpt:

```
var defaultTokens =
{ primitives: ['Infinity',
'-Infinity', 'NaN', '-NaN',
'ab','AB','000','000000'],
commands: ['all',
'bitsAllClear'] // etc. }
```

These values are drawn from our experience with testing MongoDB, but as far as the fuzzer is concerned they are just nodes of an AST, and it composes the test input from them without regard to what would be valid. Thus, our generational method produces dumbness.

It Doesn't Work Like This

We are trying to balance coverage with validation avoidance. To generate test input that has a chance of passing input validation, we could start with a template of a valid JavaScript object. The letters in this template represent placeholders:

```
{a:X, b:Y, c:Z}
```

We could then replace the capital letters with seed primitive values:

```
{a: 4294967296, b: 'ab', c:
NumberDecimal(-NaN)}
```

and replace the lowercase letters with seed MongoDB command parameters:

```
{create: 4294967296,
$add: 'ab', $max:
NumberDecimal(-NaN)}
```

This is not a valid MongoDB command, however. Even filling in a well-formatted template from a list of valid MongoDB primitives, this generated input still triggers only the validation code.

Hybrid Fuzzing

Mutational fuzzing leaves blind spots, and generational fuzzing on its own won't test interesting logic at all. When combined, however, both techniques become much more powerful. This is how our fuzzer actually works.

As it mutates existing tests, every once in a while, instead of pulling a replacement from the corpus, it generates an AST node from its list of seeds. This generational substitution reduces blind spots by producing a value not present in the corpus, while the mutational basis means the resulting command retains the structure of valid input, making it likely to pass validation. Only after it is deep in the stack does the program realize that something has gone horribly wrong. Mission accomplished.

Here is an example of hybrid fuzzing in action, using a simplified version of a test that actually exposed a bug. The fuzzer starts with the following corpus, the first line of which becomes the AST shown in Figure 4:

```
db.test.insert({x:1});
db.test.update({some:
"object"}, ...);
```

The `ObjectExpression` is converted into a placeholder node, in the same manner as mutational fuzzing.

Then the fuzzer decides that instead of replacing the placeholder node with a value from elsewhere in the corpus, it will replace it with a generated object—in this case, a `newExpression` with a large `NumberLong` as the argument, shown in Figure 5.

This yields the following test:

```
db.test.insert({a:
    new
    Number-
    Long("9223372036854775808")});
db.test.update({}, {$inc: {a: 13.0}});
```

The result is that a large 64-bit integer is inserted into MongoDB, and then its value is updated. When the ac-

Figure 4. AST of one command in the corpus. (checking on this)

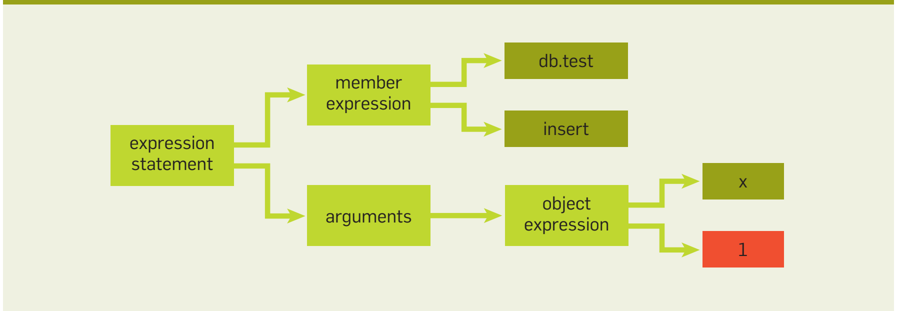
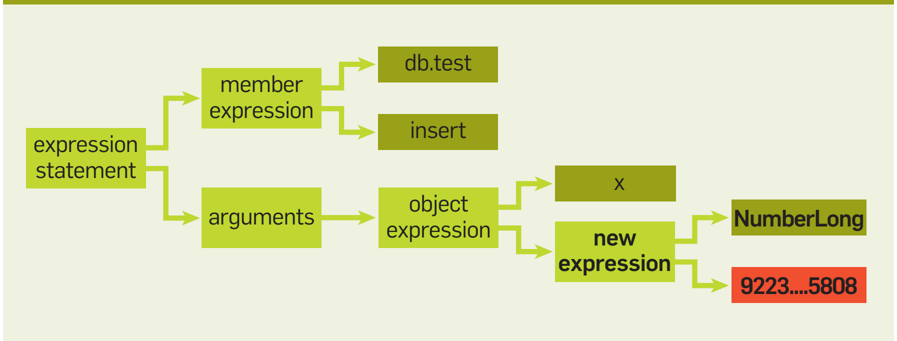


Figure 5. Placeholder replaced with a generated object.



tual test ran, it turned out that the new value would still be a large number, but not the correct one. The bug was that MongoDB stored the integer as a double internally, which has only 53 bits of precision.[13] The fuzzer was able to find this by generating the large `NumberLong`, which did not appear in any test.

The combination of mutational fuzzing with the edge cases we seed to the generational fuzzer is an order of magnitude more powerful than writing tests for these edge cases explicitly. In fact, a significant portion of the bugs the fuzzer found were triggered by values generated in this way.

An Unbridled Fuzzer Creates Too Much Noise

Ultimately, fuzz testing is a game of random numbers. Random numbers make the fuzzer powerful but can cause unforeseen problems. We needed to take some steps to ensure the fuzzer does not blow itself up. Take the following block of code, which resembles something that would be present in one of MongoDB's JavaScript tests:

```
while(coll.count() < 654321)
     assert(coll.update({a:1},
     {$set: {...}}))
```

This code does a large number of updates to a document stored in MongoDB. If we were to put it through the mutational and generational fuzzing steps described previously, the fuzzer could produce this possible test case:

```
while(true)    assert(coll.up-
date({}, {$set: {"a.654321" : 1}}))
```

The new code now tests something completely different. It tries to set the 654321st element in an array stored in all documents in some MongoDB collection.

This is an interesting test case. Using the `$set` operator with such a large array may not be something we thought of testing explicitly and could trigger a bug (in fact, it does).[12] But the interaction between the fuzzed true condition and the residual while loop is going to hang the test!—unless, that is, the assert call in the while loop fails, which could happen if the line defining `coll` in the original test (not shown here) is mutated or deleted by the fuzzer, leaving `coll` undefined. If the assert call failed, it would be caught by the Mongo shell and cause it to terminate.

Neither the hang nor the assertion failure, however, are caused by bugs in MongoDB. They are just by-products of a randomly generated test case, and they represent two classes of noise that must be filtered out of fuzz testing: branch logic and assertion failures.

Branch logic. To guard against accidental hangs, our fuzzer simply takes out all branching logic via AST manipulation. In addition to while loops, we remove `try/catch`, `break`, and `continue` statements, `do/while`, `for`, `for/in`, and `for/of` loops. These language structures are defined in a static list.

Assertion failures. For the assertion failures, every single line of generated test code is wrapped with a try/catch statement. All the logic will still be executed, but no client-side errors will propagate up and cause a failure.

After passing through this sanitizing phase, our earlier example now looks like this:

```
try {
    assert(coll.update({},
    {$set: {"a.654321" : 1}}))
} catch {}
```

So How *Does* the Fuzzer Catch Bugs?

Wrapping everything in a try/catch block keeps fuzzer-generated noise from overwhelming us with false positives, but it also prevents any bugs from surfacing through the client-side assertions our typical tests rely on. Indeed, a fuzzer has to rely on other mechanisms to detect the errors it provokes.

Tools for generic errors. The first set of tools are ones we are using anyway, for finding segmentation faults, memory leaks, and undefined behavior. Even without a fuzz tester, we would still be using these language runtime tools,[3] such as LLVM's address sanitizer[4] and undefined behavior sanitizer,[5] but they become far more useful when a fuzzer is bombarding the test target with all its random input.

These tools are good for generic coding errors, but they don't validate that a program is behaving as expected by end users. To catch issues with business logic, our fuzzer relies on assertions within the testing target that check for conditions it should not be in.

Assertions within the system under test. Many applications make liberal use of asserts to guard against illegal conditions, but fuzz testing *relies* on them to catch application logic errors. It wreaks havoc in your codebase and assumes you have instrumented your application's components such that havoc is noticed.

For example, when acting as a secondary in a MongoDB replica set, mongod has an assertion to halt if it fails to write an operation.[9] If a primary node logs a write for its secondaries, they had *better* be able to perform the write as well, or we will wind up with serious data loss when failovers happen. Since these assertions are fatal errors, the testing framework immediately notices when fuzz tests trigger them.

The limitation of randomized testing. This is really the only way that assertions can be used to catch errors provoked by randomly generated tests. Assertions in the target program can be oblivious to the tests being run; indeed, they must hold true under all circumstances (including when the program is being run by a user). In contrast, assertions within tests must be specific to the test scenario. We have already shown, however, that fuzzer-generated tests, by their nature, must not include fatal assertions. So under truly random conditions, a fuzzer will trigger *no tailored assertions*. This is a limitation of all randomized testing techniques, and it is why any good testing framework must not rely solely on randomized testing.

Triaging a Fuzzer Failure

Tests that perform random code execution and rely on target system assertions have some downsides: the problems they find have no predefined purpose; many of the operations within them might be innocuous noise; and the errors they produce are often convoluted. Failures observed at a particular line of the test might rely on a state set up by previous operations, so parts of the codebase that may be unrelated have to be examined and understood.

Thus, fuzzer failures require triage to find the smallest set of operations that trigger the problem. This can take significant human intervention, as with the known issue[17] where calling `cursor.explain()`[6] with concurrent clients causes a segmentation fault. The test that provoked this issue used a dozen clients performing different operations concurrently, so beside understanding which state the operations in the test set up, log messages from all the client and server threads had to be inspected manually and correlated with each other.

All this work is typical of triaging a fuzzer test failure, so we built a set of features that help developers sift through the chaos. These are specific to testing a MongoDB cluster across the network using JavaScript but can be used as inspiration for all fuzzing projects.

We are only interested in the lines of code that send commands to a MongoDB server, so the first step is to isolate those. Using our trusty AST manipulator, we add a print statement after every line of fuzzer code to record the time it takes to run. Lines that take a nontrivial amount of time to run typically run a command and communicate with the mongodb server. With those timers in place, our fuzz tests look like this:

```
var $startTime = Date.now();
try {
    // a fuzzer generated
    line of code
} catch (e) {
}
var $endTime = Date.now();
print('Top-level statement 0
completed in',
    $endTime - $startTime,
    'ms');

var $startTime = Date.now();
try {
    // a fuzzer generated
    line of code
} catch (e) {
}
var $endTime = Date.now();
print('Top-level statement 1
completed in',
    $endTime - $startTime,
    'ms');

// etc.
```

When we get a failure, we find the last statement that completed successfully from the log messages, and the next actual command that runs is where the triage begins.

This technique would be sufficient for identifying the trivial bugs that can cause the server to crash with one or two lines of test code. More complicated bugs require programmatic assistance to find exactly which lines of test code are causing the problem. We bisect our way toward that with a breadth-first binary search over each fuzzer-generated file. Our script recursively generates new tests containing each half of the failed code until any further removal no longer causes the test to fail.

The binary search script is not a cure-all, though. Some bugs do not reproduce consistently, or cause hangs, and require a different set of tools. The particular tools will depend entirely on your product, but one simple way to identify hangs is to use a timer. We record the runtime of a test suite, and if it takes an order of magnitude longer than the average runtime, we assume it has hung, attach a debugger, and generate a core dump.

Through the use of timers, print statements, and binary search script, we are able to triage the majority of our failures quickly and correctly. There is no panacea for debugging—every problem is new, and most require a bit of trial and error to get right. We are continuously investing in this area to speed up and simplify failure isolation.

Running the Fuzzer in the CI System

Fuzz testing is traditionally done in dedicated clusters that run periodically on select commits, but we decided to include it as a test suite in our CI framework, Evergreen. This saved us the effort of building out a new automated testing environment and saved us from dedicating resources to determine in which commit the bug was introduced.

When a fuzzer is invoked periodically, finding the offending commit requires using a tool such as git-bisect.[2] With our approach of a mutational fuzzer that runs in a CI framework, we always include newly committed tests in the corpus. Every time the fuzzer runs, we pick 150 sets of a few dozen files from the corpus at random and run each one through the fuzzer to generate 150 fuzzed files. Each set of corpus files always includes new logic added to the codebase, which means the fuzzed tests are likely testing new code as well. This is a simple and elegant way for the fuzzer to "understand" changes to the codebase without the need for significant work to parse source files or read code coverage data.

When a fuzz test causes a failure, the downstream effect is the same as any other kind of test failure, only with the extra requirement of triage.

The Fuzzer: Your Best Friend

Overall, the fuzzer has turned out to be one of the most rewarding tools in the MongoDB test infrastructure. Building off our existing suite of JavaScript tests, we were able to increase our coverage significantly with relatively little effort. Getting everything right takes time, but to get a basic barebones system started, all you need is a set of existing tests as the corpus, a syntax-tree parsing for the language of your choice, and a way to add the framework to a CI system. The bottom line is that no matter how much effort is put into testing a feature, there will inevitably be that one edge case that was not handled. In those face-palm moments, the fuzzer is there for you.

References

1. Acorn; https://github.com/ternjs/acorn.
2. Chacon, S., Straub, B. Git-bisect; https://git-scm.com/book/en/v2.
3. Clang 3.8 Documentation. Using Clang as a compiler; http://releases.llvm.org/3.8.0/tools/clang/docs/index.html#using-clang-as-a-compiler.
4. Clang 3.8 Documentation. AddressSanitizer; http://releases.llvm.org/3.8.0/tools/clang/docs/AddressSanitizer.html.
5. Clang 3.8 Documentation. UndefinedBehaviorSanitizer; http://releases.llvm.org/3.8.0/tools/clang/docs/UndefinedBehaviorSanitizer.html.
6. Cursor.explain(). MongoDB Documentation; https://docs.mongodb.com/manual/reference/method/cursor.explain/.
7. Déjà vu Security. Generation fuzzing. Peach Fuzzer, 2014; http://community.peachfuzzer.com/GenerationMutationFuzzing.html.
8. Erf, K. Evergreen continuous integration: Why we reinvented the wheel. MongoDB Engineering Journal 2016; https://engineering.mongodb.com/post/evergreen-continuous-integration-why-we-reinvented-the-wheel/.
9. GitHub. MongoDB; https://github.com/mongodb/mongo/blob/f5c9d27ca6f0f4e1e2673c64b84b628ac29493ec/src/mongo/db/repl/sync_tail.cpp#L1042.
10. Godefroid, P., Levin, M.Y., Molnar, D. SAGE: Whitebox fuzzing for security testing. *Commun. ACM 55*, 3 (Mar. 2012, 40-44; http://courses.cs.washington.edu/courses/cse484/14au/reading/sage-cacm-2012.pdf.
11. Guo, R. Mongos segfault when invoking .explain() on certain operations. MongoDB, 2016; https://jira.mongodb.org/browse/SERVER-22767.
12. Guo, R. $push to a large array fasserts on secondaries. MongoDB, 2016; https://jira.mongodb.org/browse/SERVER-22635.
13. Kamsky, A. Update considers a change in numerical type to be a noop. MongoDB, 2016; https://jira.mongodb.org/browse/SERVER-16801.
14. McCloskey, B., et al. Parser API. Mozilla Developer Network, 2015; https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey/Parser_API#Expressions.
15. Nossum, V., Casasnovas, Q. Filesystem fuzzing with American Fuzzy Lop. Oracle Linux and VM Development—Ksplice Team, 2016; https://events.linuxfoundation.org/sites/events/files/slides/AFL filesystem fuzzing, Vault 2016_0.pdf.
16. Ruderman, J. Introducing jsfunfuzz. Indistinguishable from Jesse, 2007; https://www.squarefree.com/2007/08/02/introducing-jsfunfuzz/.
17. Siu, I. Explain("executionStats") can attempt to access a collection after it has been dropped. MongoDB, 2016; https://jira.mongodb.org/browse/SERVER-24755.
18. Storch, D. MongoDB, jstests. GitHub, 2016; https://github.com/mongodb/mongo/tree/r3.3.12/jstests.

Robert Guo is a software engineer on the MongoDB server team, focusing on data consistency and correctness. He is currently working on MongoDB's JavaScript fuzzer.

DOI:10.1145/3024928

Article development led by acmqueue
queue.acm.org

Expert-curated guides to the best of CS research.

Research for Practice: Cryptocurrencies, Blockchains, and Smart Contracts; Hardware for Deep Learning

OUR FOURTH INSTALLMENT of Research for Practice covers two of the hottest topics in computer science research and practice: cryptocurrencies and deep learning.

First, **Arvind Narayanan** and **Andrew Miller**, co-authors of the increasingly popular open access Bitcoin textbook, provide an overview of ongoing research in cryptocurrencies. This is a topic with a long history in the academic literature that has recently come to prominence with the rise of Bitcoin, blockchains, and similar implementations of advanced, decentralized protocols. These developments—and colorful exploits such as the DAO vulnerability in June 2016—have captured the public imagination and the eye of the popular press. In the meantime, academics have been busy, delivering new results in maintaining anonymity, ensuring usability, detecting errors, and reasoning about decentralized markets, all through the lens of these modern cryptocurrency systems. It is a pleasure having two academic experts deliver the latest updates from the burgeoning body of academic research on this subject.

Next, **Song Han** provides an overview of hardware trends related to another long-studied academic problem that has recently seen an explosion in popularity: deep learning. Fueled by large amounts of training data and inexpensive parallel and scale-out compute, deep-learning-model architectures have seen a massive resurgence of interest based on their excellent performance on traditionally difficult tasks such as image recognition. These deep networks are compute-intensive to train and evaluate, and many of the best minds in computer systems (for example, the team that developed MapReduce) and AI are working to improve them. As a result, Song has provided a fantastic overview of recent advances devoted to using hardware and hardware-aware techniques to compress networks, improve their performance, and reduce their often large amounts of energy consumption.

As always, our goal in this column is to allow our readers to become experts in the latest topics in computer science research in a weekend afternoon's worth of reading. To facilitate this process, as always, we have provided open access to the ACM Digital Library for the relevant citations from these selections so you can read the research results in full. Please enjoy!

—Peter Bailis

Peter Bailis is an assistant professor of computer science at Stanford University. His research in the Future Data Systems group (futuredata.stanford.edu/) focuses on the design and implementation of next-generation data-intensive systems.

Cryptocurrencies, Blockchains, and Smart Contracts

By Arvind Narayanan and Andrew Miller

Research into cryptocurrencies has a decades-long pedigree in academia, but decentralized cryptocurrencies (starting with Bitcoin in 2009) have taken the world by storm. Aside from being a payment mechanism "native to the Internet," the underlying blockchain technology is touted as a way to store and transact everything from property records to certificates for art and jewelry. Much of this innovation happens in the broader hobbyist and entrepreneurial communities (with increasing interest from established industry players); Bitcoin itself came from outside academia. Researchers, however, have embraced cryptocurrencies with gusto and have contributed important insights.

Here we have selected three prominent areas of inquiry from this young field. Our selections of research papers within each area focus on relevance to practitioners and avoid such areas as scalability that are of interest primarily to cryptocurrency designers. Overall, the research not only exposes important limitations and pitfalls of the technology, but also suggests ways to overcome them.

Anonymity, Privacy, and Confidentiality

Meiklejohn, S. et al.
A fistful of Bitcoins: Characterizing payments among men with no names. In *Proceedings of the Internet Measurement Conference, 2012, 127–140.* https://www.usenix.org/system/files/login/articles/03_meiklejohn-online.pdf.

Bitcoin exists in a state of tension between anonymity (in the sense that real identities are not required to use the system) and traceability (in that all transactions are recorded on the blockchain, which is a public, immutable, and global ledger). In practice, the privacy of vanilla Bitcoin comes from obscurity: users may create as many addresses as they like and shuffle their coins around, even creating a new address for each transaction. But this paper demonstrates that "address clustering" can be very effective, applying a combination of heuristics to link together all the pseudo-identities controlled by an individual or entity.

Anonymity in cryptocurrencies is a matter of not just personal privacy, but also confidentiality for enterprises. Given advanced transaction graph analysis techniques, without precautions, the blockchain could easily reveal cash flow and other financial details.

Sasson, E.B. et al.
Zerocash: Decentralized anonymous payments from Bitcoin. In *Proceedings of the IEEE Symposium on Security and Privacy, 2014.* http://zerocash-project.org/media/pdf/zerocash-extended-20140518.pdf.

There are many different proposals for improving the privacy of cryptocurrencies. These range from Bitcoin-compatible methods of "mixing" (or "joining") coins with each other, to designs for entirely new cryptocurrency protocols that build in privacy from the beginning. Perhaps the most radical proposal is Zerocash, an alternative cryptocurrency design that uses cutting-edge cryptography to hide all information from the blockchain except for the existence of transactions; each transaction is accompanied by a cryptographic, publicly verifiable proof of its own validity. Roughly, the proof ensures that the amount being spent is no more than the amount available to spend from that address. The paper is long and intricate, and the underlying mathematical assumptions are fairly new by cryptographic standards. But this fact itself is food for thought: to what extent does the security of a cryptocurrency depend on the ability to comprehend its workings?

Endpoint Security

The Achilles' heel of cryptocurrencies has been the security of endpoints, or the devices that store the private keys that control one's coins. The cryptocurrency ecosystem has been plagued by thefts and losses resulting from lost devices, corrupted hard drives, malware, and targeted intrusions. Unlike fiat currencies, cryptocurrency theft is instantaneous, irreversible, and typically anonymous.

Bitcoin itself came from outside academia. Researchers, however, have embraced cryptocurrencies with gusto and have contributed important insights.

> **Prediction markets allow market participants to trade shares in future events (such as "Will the U.K. initiate withdrawal from the E.U. in the next year?") and turn a profit from accurate predictions.**

Eskandari, S., Barrera, D., Stobert, E., Clark, J.
A first look at the usability of Bitcoin key management. *Workshop on Usable Security, 2015.*
http://users.encs.concordia.ca/~clark/papers/2015_usec.pdf.

This paper studies six different ways to store and protect one's keys, and evaluates them on 10 different criteria encompassing security, usability, and deployability. No solution fares strictly better than the rest. Users may benefit considerably from outsourcing the custody of their keys to hosted wallets, which sets up a tension with Bitcoin's decentralized ethos. Turning to Bitcoin clients and tools, the authors find problems with the metaphors and abstractions that they use. This is a ripe area for research and deployment, and innovation in usable key management will have benefits far beyond the world of cryptocurrencies.

Smart Contracts

One of the hottest areas within cryptocurrencies, so-called smart contracts, are agreements between two or more parties that can be automatically enforced without the need for an intermediary. For example, a vending machine can be seen as a smart contract that enforces the rule that an item will be dispensed if and only if suitable coins are deposited. Today's leading smart-contract platform is called Ethereum, whose blockchain stores long-lived programs, called contracts, and their associated state, which includes both data and currency. These programs are immutable just as data on the blockchain is, and users may interact with them with the guarantee that the program will execute exactly as specified. For example, a smart contract may promise a reward to anyone who writes two integers into the blockchain whose product is RSA-2048—a self-enforcing factorization bounty!

Luu, L., Chu, D-H., Olickel, H., Saxena, P., Hobor, A.
Making smart contracts smarter.
In *Proceedings of ACM SIGSAC Conference on Computer and Communications Security, 2016, 254–269.*
https://dl.acm.org/citation.cfm?id=2978309.

Unfortunately, expressive programming languages are difficult to reason about. An ambitious smart contract called The DAO suffered a theft of an estimated $50 million thanks to a litany of security problems. (Ultimately, this theft was reversed by a networkwide "hard-fork" upgrade.) The authors study four classes of security vulnerabilities in Ethereum smart contracts, and build a tool to detect them based on a formalization of Ethereum's operational semantics. They find that thousands of contracts on the blockchain are potentially vulnerable to these bugs.

Clark, J., Bonneau, J., Felten, E.W., Kroll, J.A., Miller, A. and Narayanan, A.
On decentralizing prediction markets and order books. *Workshop on the Economics of Information Security, State College, PA, 2014.*
http://www.econinfosec.org/archive/weis2014/papers/Clark-WEIS2014.pdf.

If smart-contract technology can overcome these hiccups, it could enable decentralized commerce—that is, various sorts of markets without intermediaries controlling them. This paper studies how one type of market—namely, a prediction market—could be decentralized. Prediction markets allow market participants to trade shares in future events (such as "Will the U.K. initiate withdrawal from the E.U. in the next year?") and turn a profit from accurate predictions. In this context the authors grapple with various solutions to a prominent limitation of smart contracts: they can access only data that is on the blockchain, but most interesting data lives outside it. The paper also studies decentralized order books, another ingredient of decentralized markets.

Overcoming the Pitfalls

Cryptocurrencies implement many important ideas: digital payments with no central authority, immutable global ledgers, and long-running programs that have a form of agency and wield money. These ideas are novel, yet based on sound principles. Entrepreneurs, activists, and researchers have envisioned many powerful applications of this technology, but predictions of a swift revolution have so far proved unfounded. Instead, the community has begun the long, hard work of integrating the technology into Internet infrastructure and existing institutions. As we have seen, there are pitfalls for the unwary in using and applying cryptocurrencies: privacy, security, and interfacing with the real world. These will be fertile areas of research and development in the years to come.

Arvind Narayanan is an assistant professor of computer science at Princeton, where he leads a research team investigating the security, anonymity, and stability of cryptocurrencies as well as novel applications of blockchains. He also leads the Princeton Web Transparency and Accountability Project, to uncover how companies collect and use our personal information.

Andrew Miller is an assistant professor in Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. He is an associate director of the Initiative for Cryptocurrencies and Contracts (IC3) at Cornell and an advisor to the Zcash project.

Hardware For Deep Learning

By Song Han

Deep neural networks (DNNs) have evolved to a state-of-the-art technique for machine-learning tasks ranging from computer vision to speech recognition to natural language processing. Deep-learning algorithms, however, are both computationally and memory intensive, making them power-hungry to deploy on embedded systems. Running deep-learning algorithms in real time at subwatt power consumption would be ideal in embedded devices, but general-purpose hardware is not providing satisfying energy efficiency to deploy such a DNN. The three papers presented here suggest ways to solve this problem with specialized hardware.

The Compressed Model

Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M.A., Dally, W.J.
EIE: Efficient inference engine on compressed deep neural network. In *Proceedings of the International Symposium on Computer Architecture, 2016.*
https://arxiv.org/pdf/1602.01528v2.pdf.

This work is a combination of algorithm optimization and hardware specialization. EIE (efficient inference engine) starts with a deep-learning-model compression algorithm that first prunes neural networks by 9–13 times without hurting accuracy, which leads to both computation saving and memory saving; next, using pruning plus weight sharing and Huffman coding, EIE further compresses the network 35–49 times, again without hurting accuracy. On top of the compression algorithm, EIE is a hardware accelerator that works directly on the compressed model and solves the problem of irregular computation patterns (sparsity and indirection) brought about by the compression algorithm. EIE efficiently parallelizes the compressed model onto multiple processing elements and proposes an efficient way of partitioning and load balancing both the storage and the computation. This achieves a speedup of 189/13 times and an energy efficiency improvement of 24,000/3,400 times over a modern CPU/GPU.

Optimized Dataflow

Chen, Y.-H., Emer, J., Sze, V.
Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In *Proceedings of the International Symposium on Computer Architecture, 2016.* https://www.researchgate.net/publication/301891800_Eyeriss_A_Spatial_Architecture_for_Energy-Efficient_Dataflow_for_Convolutional_Neural_Networks.

Deep-learning algorithms are memory intensive, and accessing memory consumes energy more than two orders of magnitude more than ALU (arithmetic logic unit) operations. Thus, it's critical to develop dataflow that can reduce memory reference. Eyeriss presents a novel dataflow called RS (row-stationary) that minimizes data-movement energy consumption on a spatial architecture. This is realized by exploiting local data reuse of filter weights and feature map pixels (that is, activations) in the high-dimensional convolutions, and by minimizing data movement of partial sum accumulations. Unlike dataflows used in existing designs, which reduce only certain types of data movement, the proposed RS dataflow can adapt to different CNN (convolutional neural network) shape configurations and reduce all types of data movement through maximum use of PE (processing engine) local storage, direct inter-PE communication, and spatial parallelism.

Small-Footprint Accelerator

Chen, T., Wang, J., Du, Z., Wu, C., Sun, N., Chen, Y., Temam, O.
DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems, 2014.*
http://pages.saclay.inria.fr/olivier.temam/files/eval/CDSWWCT14.pdf.

Recent state-of-the-art CNNs and DNNs are characterized by their large sizes. With layers of thousands of neurons and millions of synapses, they place a special emphasis on interactions with memory. DianNao is an accelerator for large-scale CNNs and DNNs, with a special emphasis on the impact of memory on accelerator design, performance, and energy. It takes advantage of dedicated storage, which is key for achieving good performance and power. By carefully exploiting the locality properties of neural network models, and by introducing storage structures custom designed to take advantage of these properties, DianNao shows it is possible to design a machine-learning accelerator capable of high performance in a very small footprint. It is possible to achieve a speedup of 117.87 times and an energy reduction of 21.08 times over a 128-bit 2GHz SIMD (single instruction, multiple data) core with a normal cache hierarchy.

Looking Forward

Specialized hardware will be a key solution to make deep-learning algorithms faster and more energy efficient. Reducing memory footprint is the most critical issue. The papers presented here demonstrate three ways to solve this problem: optimize both algorithm and hardware and accelerate the compressed model; use an optimized dataflow to schedule the data movements; and design dedicated memory buffers for the weights, input activations, and output activations. We can look forward to seeing more artificial intelligence applications benefit from such hardware optimizations, putting AI everywhere, in every device in our lives.

Song Han is a Ph.D. student at Stanford University, Stanford, CA. He proposed deep compression that can compress state-of-the art CNNs by 10–49 times and designed EIE (efficient inference engine), a hardware architecture that does inference directly on the compressed sparse model.

DOI:10.1145/2996181

User attitudes toward online intellectual property reveal how far social norms have strayed from legal notions of ownership.

BY CATHERINE C. MARSHALL AND FRANK M. SHIPMAN

Who Owns the Social Web?

USER-CONTRIBUTED CONTENT plays an increasingly important role in the Internet's evolution, overtaking professionally created and curated resources. Sophisticated recording technologies allow non-professionals to produce high-quality photos and videos. Improved editing and sharing applications facilitate other aspects of media creation, including larger-scale collaborative efforts. And social media venues give their users new opportunities to publish, curate, and recommend content. Every phase of the creative process—from recording to editing to publishing—has become more popular and interactive. At the same time, content ownership has become more complicated. Any distinct item may be associated with a virtual web of stakeholders.

» key insights

- Intellectual property law and social norms concerning content ownership are diverging in conspicuous ways; we find that legally contentious actions (such as downloading and saving content) may seem benign to most Internet users.
- Managing rights relies on content owners' ability to envision plausible reuse scenarios, including commercial reuse of their content as data, and predicting which are most likely.
- Everyday reuse of social media content is opportunistic, pragmatic, and highly contextual; users reason about the fairness of reusing other people's content but do not necessarily trust them to do the same.

contributed articles
User attitudes toward online intellectual property reveal how far social norms have strayed from legal notions of ownership.
BY CATHERINE C. MARSHALL AND FRANK M. SHIPMAN
Who Owns
Social
?
key insights
IMAGE REMOVED

ILLUSTRATION BY JUSTIN METZ

A product review posted on Amazon might attract hundreds of comments that contribute substantively to the review's value and credibility. Videos on YouTube might respond to, excerpt, or satirize one another. Ongoing conversational threads on Twitter are held together by hashtags and @responses. Gamers use their avatars to interact with one another against the backdrop of a virtual universe and, in so doing, create new forms of data that build on the game's commercial content. Moreover, as individuals develop rich personal profiles, they publish new kinds of online representations of themselves.

This complex non-professional digital-media landscape, along with newfound opportunities for copying, excerpting, and remixing professionally produced media, poses new challenges for managing intellectual property.

Many social, legal, and technological forces shape our perceptions of who can do what with Internet content. As law professor and social activist Lawrence Lessig points out, in addition to legal notions of copyright, market forces and technologically enforced prohibitions constrain users' actions; additionally, emerging social norms make some online user behaviors seem acceptable to most people, while other behaviors are perceived as reprehensible.[6]

In *Order Without Law*, property law scholar Robert C. Ellickson demonstrated how people settle their disputes and regulate their behavior via these social norms; his analysis shows the importance of the norms and how they can be as effective as law.[3] Legal scholar John Tehranian has highlighted how ordinary people (rather than legal scholars or jurists) now have a heightened awareness of the issues surrounding content ownership and, at the same time, the gap between the body of copyright legislation and the social norms that govern ownership behavior is growing.[22] That is, although people are aware there are legal prescriptions for ownership and its reach, they are guided instead by social norms. By reflecting on his own online practices, Tehranian even showed how absurd copyright law has become relative to more sensible social norms.

In this article, we take a bottom-up view of content ownership and control, seeking to identify the norms and practices of everyday media users. Since 2010, we have conducted a series of surveys aiming to discover the social norms associated with content ownership and control and identify which media-specific user behaviors shape them. Once we arrive at a rich characterization of these norms, policy and technology can be designed accordingly and conflicts between practice and design can be anticipated and addressed when unavoidable.

Uncovering Social Norms

Having acknowledged the role of social norms in reflecting and guiding online behavior, it seems like we might want to know what these social norms are. How situational are they? How much do they vary across media types or services?

To examine them, we must identify people's current practice (what they do), both as media consumers and as media creators, and we need to get them to articulate their aspirations. What do they think is fair? What should they do? What should others do? What behavior do they object to? It is important to distinguish practice from aspiration; in a pinch or when no one is looking, people's behavior is highly situated and unselfconscious. It is not so much that study-participants might want to mislead us, but more that they are not always aware of what they usually do.

When we began these studies, we thus sought to pin down both practice and aspiration among a broad set of people who spend a substantial amount of time online and use specific types of social media services. We designed and ran a set of eight studies over the next four years, each focusing on the ownership and control of a different media type and service: tweets,[9] photos,[10] reviews,[20] podcasts, recorded videoconferences and educational videos,[12] recordings from multiplayer online games,[21] and Facebook content.[13]

We screened participants for familiarity with the content type or social media service in question, then checked responses according to documented criteria to ensure the study had been completed in good faith, a process detailed in articles covering the individual studies.[9,10,12,13,20,21] We collected valid responses from a total of 1,738 participants. Many had attended college; approximately one-third were students at the time of the study. Most were between 18 and 40, although close to 20% were over 40 at the time of the study. Participants' individual interests and online activities varied, and they used a range of applications and services in addition to the one under investigation. Participants were generally both content consumers and creators. Two-thirds of them in the six studies after 2012 had been on the Internet more than 10 years. The accompanying table summarizes key participant demographics.

In the studies, we adapted a technique that has been used successfully in legal education[8] and legal argument analysis:[16] scenarios (or cases) plus hypotheticals. Hypotheticals explore the features of a heuristic or rule, with the aim of discovering the edges of how and when it applies. For example, we might want to know if it is okay for a user to download and store a stranger's photo if the user is the subject of the photo. To explore these rights, consider a scenario in which Sophia, a 25-year-old woman, encounters a photo of herself on Instagram that was taken at a wedding reception, and she does not know the amateur photographer who was sitting at her table and took the picture. Sophia likes the candid shot and wants to download it and store it locally. Hypotheticals can then be used to explore Sophia's rights to the photo by varying different features, say, the action she is taking with the photo, storing it rather than re-uploading and using it as her Facebook profile photo; her relationship to the photo's subject—herself—rather than if the subject was, say, another wedding guest; the status of the photographer, a fellow wedding guest, not a professional photographer; and where Sophia is keeping the photo, saving it to her hard drive rather than storing it on a cloud-storage service. Scenarios thus situate the hypotheticals in a story with concrete details derived from real-life situations.

The scenarios and hypotheticals are both more engaging than an abstract version of the question (such as "Should you be allowed to down-

Key participant demographics. Survey topics are listed in the order the studies were conducted. Values in the "Number of participants" column are the number of responses retained after screening. Parenthetical values are the total number of responses collected.

Study topic	Number of participants	Percent attended college	Percent current students	Percent female	Percent born after 1980	Percent with >10 years Internet experience
Tweets	173 (190)	88%	NA	61%	75%	49%
Photos	242 (250)	91%	34%	71%	66%	55%
Reviews	203 (216)	92%	32%	59%	61%	66%
Podcasts	225 (239)	90%	31%	44%	69%	67%
Videos	200 (228)	93%	24%	47%	60%	67%
Educational videos	209 (250)	94%	36%	50%	66%	67%
Multiuser games	241 (251)	80%	29%	35%	77%	66%
Facebook content	244 (250)	92%	25%	45%	72%	68%
Total	1,737 (1,874)	90%	NA	51%	68%	63%

load any picture of yourself and save it?") and less apt to leave the details to chance (such as "Are we asking about a posed picture or a candid picture?" and "Is the photographer a professional using a camera or a fellow party guest using a smartphone?"). These details can make a difference in how the hypotheticals are interpreted and what response(s) they might elicit.

We captured participant reactions using a seven-point Likert scale, from strongly disagree to strongly agree. Each study presented from 16 to 28 hypotheticals associated with from two to four related scenarios. We discussed methodological details and aggregate participant demographics for the first six studies in an earlier paper.[11]

The studies explored three types of common user actions: saving, sharing, and removing:

Saving. We defined saving as the act of intentionally downloading content from a social media site or service and storing it to a place under the user's control. Saving user-contributed media has minimal effect on others; it neither affects the digital original nor will most legal copyright holders ever know the content has been saved elsewhere;

Sharing. We defined sharing as reposting existing user-contributed media on another site or service, possibly without attribution, along with varying degrees of content transformation and varying user intent. In essence, sharing tests the social norms that circumscribe the limits of fair use. Our first three studies—covering tweets, photos, and reviews—distinguished between sharing by, say, posting content on one's Facebook account so it can be accessed by a limited social group and publishing openly as, say, a public blog post, but participants did not seem to notice this distinction themselves without considerable explanation. Our later studies—covering podcasts, recorded videoconferences, educational videos, recordings from multiplayer online games, and Facebook content—did not make this distinction; and

Removing. We defined removing as deleting or limiting access to user-contributed content. Removal is an action that tests the limits of media ownership and control, since it is not usually supported if the remover is not the explicit content owner. Because removal in practice is often provoked by a desire to curate one's digital footprint or reflect a changing notion of privacy, we designed the removal scenarios to elicit feelings of shared ownership, as in "You have posted something I feel I should be able to control, as with, say, a picture of me.

We returned to saving, sharing, and removing throughout the studies.

Is saving someone else's photo appreciably different from saving someone else's tweet?

Highlights of Study Results

Our eight media-specific studies explored features that influence people's attitudes about the ownership and control of user-contributed content. Using a consistent set of actions—saving, sharing, and removing—supported comparing whether ownership rights are sensitive to expectations introduced by media type and nature of the actions users take upon them. That is, is saving someone else's photo appreciably different from saving someone else's tweet? Moreover, the scenarios helped us explore media-independent features, including:

Original context. This feature tests whether the content's original context influenced our study participants' perceptions of ownership rights; for example, do users have the same right to save a photo of a vintage picnic table they encounter on another user's public Flickr account as they do a similar photo that was used as an eBay product description?;

User's relationship to content. This feature tests some of the complexities of ownership. For example, if a person is the subject of a photo, as opposed to being the photographer, should this particular fact influence the person's right to save, reuse, or remove the photo from the service where it resides?;

Commercial concerns. This feature considers users' understanding of corporate ownership rights, as well as commercial use by individuals, apart from any terms and conditions spelled out by the service. For example, does the service owner have the right to save private communication that occurs within the service? And does it have the right to analyze the public communication it supports? Does it have the right to remove content it deems offensive?;

Genre-derived properties. Some content genres may have properties that raise specific expectations about associated rights. For example, media considered ephemeral (such as an in-game chat session or other forms of communication) may influence perceptions of another user's right to save the content, regardless of whether that user was a participant in the conversation; and

Disaggregation. Disaggregation tests whether the rights to constituent parts of an item are different from the rights to the whole. This feature has allowed us to test whether, for example, the audio track of a recorded video inherits ownership rights from the video.

Here, we discuss the highlights of our findings, including social norms that emerged across studies and sometimes across actions. We also note media-specific norms and where norms break down.

Saving Social Media

To our participants, saving is the most benign, or least ethically contentious, action the scenarios explored. In the surveys, we define saving as an intentional act of downloading something—a photo, podcast, document, or video—to user-controlled storage to maintain a copy, rather than a side effect of performing some other action (such as viewing a webpage).

Users may save content (such as a tweet or a photo) because they fear its owner will delete it, because the site itself offers no guarantee of permanence (such as a story in a newsfeed may disappear and be difficult to re-find), or simply because they want to have a copy on hand. In the scenarios we spelled out in the surveys, saving was always motivated so participants would not imagine differing reasons for saving something; for example, guests interviewed on a podcast might want to save copies of the podcast for themselves.

The scenarios distinguished between saving for permanence and saving for reuse; the surveys considered reuse separately and are discussed in the next section. The scenarios also posited that the person was saving content without impediment; no tricks (such as screen captures of a Snapchat session) or special knowledge were necessary. That way, saving would not seem contrary to the media creator's expectations. In addition to testing the features outlined earlier, the hypotheticals checked two other aspects of saving—saving to cloud storage and explicitly imposed limits on saving.

Cloud storage. Cloud storage is often portrayed in the popular media and in user interfaces as a seamless extension of local storage. Yet it is never fully under user control, and service-provider terms and conditions may apply. From a rights perspective, is saving downloaded content to local storage (such as on the person's hard drive) different from saving it to private cloud storage (such as in the person's Dropbox folder)?; and

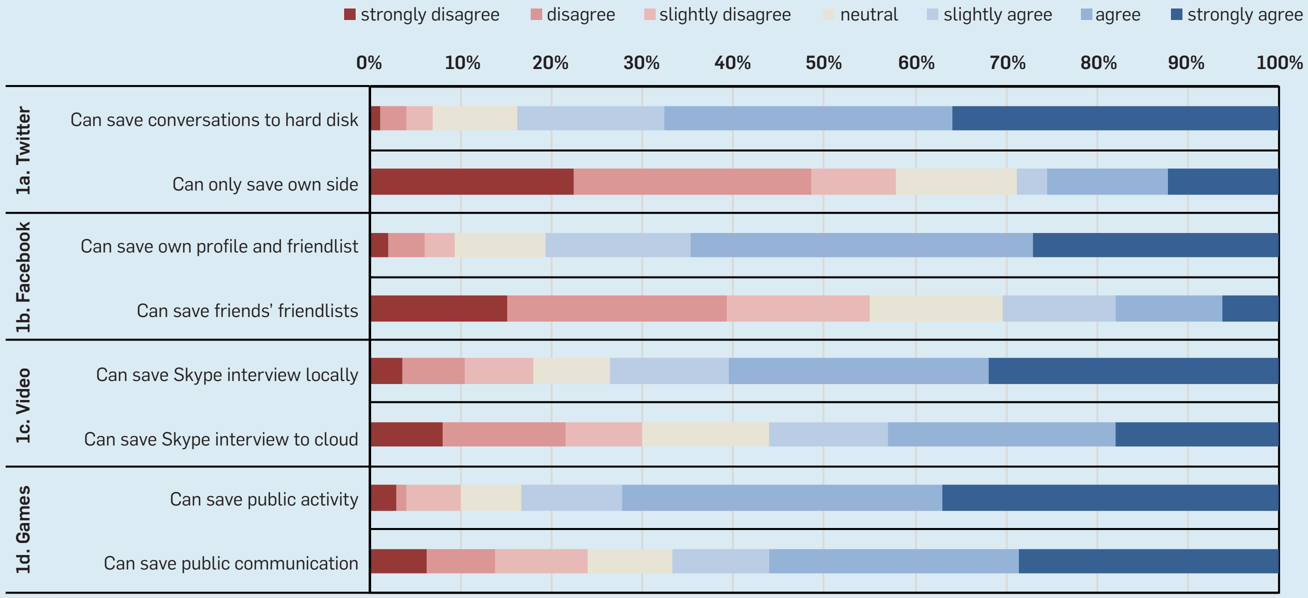
Limits. Responses to hypotheticals in the surveys suggest people expect to be able to download much of what they encounter online. This baseline may be tested by imposing artificial limits. For example, suppose people are permitted to save tweets they authored themselves but not the tweets other users wrote in response?

Our results have confirmed the baseline condition that participants usually felt individuals should be able to save anything they encounter on the Internet to local storage, regardless of whether the content is published on the open web or shared on a social media service, as long as the content is public.

This result was reaffirmed by our own hypotheticals that tested the idea of imposing limits on saving; these limits were based on a strong interpretation of ownership rights. Participants

Figure 1. Social norms for saving online content.

Pair 1a compares saving all parts of a Twitter conversation with a conventional ownership limit imposed on saving tweets, thus saving only one's own tweets. Pair 1b shows the effect of social distance on saving Facebook content. Pair 1c compares saving content locally with saving content to the cloud. And Pair 1d compares saving publicly visible in-game activity—on-screen avatar presence and action—with saving public in-game communication.



often disagreed with these imposed limits. In an extreme case—a hypothetical that limited saving tweets to saving only one's own tweets—58% (100/173) of the participants disagreed at least somewhat. Figure 1a contrasts saving all tweets in a Twitter conversation and saving just one's own tweets. That is, participants like the way content is controlled now; for example, if users are downloading content and saving it to local storage, they should not be limited to just the content they clearly own (such as photographs they have taken and posted, bon mots they have typed, or their own side of a conversation); instead, our study participants feel the norm is unfettered saving.

There are exceptions to this rule that also characterize norms associated with saving content. The strongest effect stems from the introduction of social networks. Participants respect explicit boundaries set by their social connections. While it seems perfectly acceptable to save any content encountered on the open Web, once one is inside Facebook, for example, different rules seem to apply. Our survey participants expressed a strong negative reaction to the hypothetical that one has the right to save the profile of a friend of a friend, even given reasonable motivation for doing so. Figure 1b contrasts one's right to save one's own profile and friends list (197/244, or 81%, agreed) with the right to save the equivalent content for one's friends (only 74/244, or 30%, agreed). This reaction is surprising, given the laissez-faire attitude about saving in general.

Two weaker effects also appeared in our survey results. First, saving to the cloud is viewed differently from saving to local storage. As a test, study participants judged two hypotheticals that differed only in where the downloaded content was stored. In the first, a recorded job interview—a Skype-based video—was stored to the user's local hard drive and in the second to a cloud storage service. In the first hypothetical, 18% disagreed with an individual's right to record the video and save it; in the second, the disagreement jumped to 30% (see Figure 1c.) Study participants may feel local storage is more private than cloud storage or are perhaps concerned that terms and conditions give one less control over the ultimate disposition of the content. This effect may diminish as people become accustomed to cloud storage but may also grow if privacy breaches continue to be reported in the news.

A second effect stems from users' expectations that certain media types associated with communication will remain ephemeral. Some of our study participants were uncomfortable with the idea that a conversation may be recorded and stored locally for an unspecified period, even without intimations of reuse. Hypotheticals in a multiplayer-gaming scenario in one of our surveys revealed that participants were generally undisturbed by the thought of players saving recordings of other players' public avatar appearance, gestures, and other movement we refer to as "activity"; only 24/241, or 10%, disagreed with the right to save this content. Comparable recordings of public in-game conversations were regarded more skeptically; 58/241, or 24%, disagreed with the right to save this content. Communication carried on in public still carries with it an expectation of ephemerality that could change as standards for recording others in public grow increasingly lax.

Reusing Social Media

Reuse is one of the more contentious aspects of current legal interpretation of copyright and fair use. Some major social media sites (such as YouTube) receive numerous take-down notices for content that copyright owners feel has been inappropriately reused or reposted. Meanwhile, as Tehranian predicted in 2007,[22] nuanced social norms have evolved to handle reuse of different media types and genres in a variety of circumstances.

Our reuse scenarios and hypotheticals examined at least eight features: the five described earlier—original context, the user's relationship to the content, commercial concerns, genre-derived properties, and disaggregation of constituent content—plus three additional concepts salient to reuse:

Public good. Public good scenarios seek a balance between individual rights (such as to privacy and to be forgotten) and the countervailing public interest (such as the right to preserve, access, and reuse historical content). Each of our studies included an institutional-archiving scenario that posits the creation of a media-type-specific collection (such as an archive of public Facebook content or YouTube videos); associated hypotheticals in our surveys tested varying limits on access;

Permission. In our early studies, open-ended questions revealed that some participants thought permission was the essential bridge to fair use, although a legal approach to fair use does not require one to seek or obtain permission. Our later studies tested the mitigating force of permission with hypotheticals; that is, if permission is sought or obtained, does it drastically change participants' attitudes about reuse?; and

Venue and purpose. Our surveys used hypotheticals to compare reuse of the same content in varying contexts. For example, do participants' attitudes change if an Amazon book review is republished on a blog, on Facebook, or on another online bookstore? Because purpose may be entwined with venue, hypotheticals we spelled out in our studies specified a similar purpose so participants would judge them against the same baseline. Of special note are the hypotheticals in which user-contributed content is reused as data. This practice is common, as personal information is analyzed to draw conclusions about users as a group or to target advertising. Reuse hypotheticals also distinguished between a positive or neutral purpose and a distinctly negative purpose.

Our study results confirm the widely held user expectation that attitudes toward reuse crucially depend on circumstances and may stray far from what is legally permissible under systematized U.S. fair-use provisions.[17] Aufdeheide et al.'s work with journalists[1] shows that users' stated attitudes are often more conservative than the law dictates, not less. Nonetheless, our participants' attitudes also confirmed Fiesler's and Bruckman's observation[4] of the emergence of a rich set of reuse heuristics, norms, and self-policing tactics within communities, as reuse becomes not only commonplace but lauded in the creative arena.[5]

One of the more notable of our results is derived from commercial reuse hypotheticals, especially as they propose social media content that is reused as big data. Big data is often used in aggregate to profile community behavior and individually in personalization mechanisms. The strongest disagreement arose when we asked the participants to react to a hypothetical in which Facebook sold users' profile information to Amazon (to target advertising). Figure 2a shows that more than 84% of our survey's 244 participants disagreed with the premise that this reuse is within the service's rights, and 57% disagreed vehemently. This was the most contentious of all of the reuse hypotheticals. Yet the same basic hypothetical was palatable (less than 7% objected) if the account owner's permission was solicited. The two hypotheticals define opposite ends of a spectrum of reuse attitudes.

To help tease apart the effects of the different concepts—commercial reuse, selling data, and permission—our survey proposed a third hypothetical—that Facebook can analyze internal communication among users to target advertising. This hypothetical elicited a strongly negative reaction (over 65% of our survey's 244 participants were at least somewhat negative), although their reaction was milder than the reaction to the initial hypothetical—commercial reuse of personal content—described earlier (see Figure 2a).

Commercial reuse tends to be regarded by our study participants in a cynical light. Even fairly justifiable commercial reuse—an author reusing a reader's positive review on the author's own website—elicited negative reactions from approximately one-third of survey participants (68 of 203 responses). Yet participants recognized the value of their personal data and exhibited a willingness to monetize it themselves; for example, a hypothetical that posited the sale of one's own Facebook data for personal gain tested relatively positively; 59%, or 145 of 244 participants, agreed they should be able to sell their personal content themselves.

Reusing information in a way that changes the information's veracity (that is, so it becomes deceptive, false, or is recast in an unintended way) elicited significantly more disapproval than benign reuse. Contrast the negative responses to a hypothetical in which a podcast guest re-edited a recording of himself and vetted audience comments to eliminate damaging, but valid, material (57% of 225 participants disagreed with the guest's right to create this remix) with a comparable positive response to republication of the podcast to seek a broader audience (81% favorable).

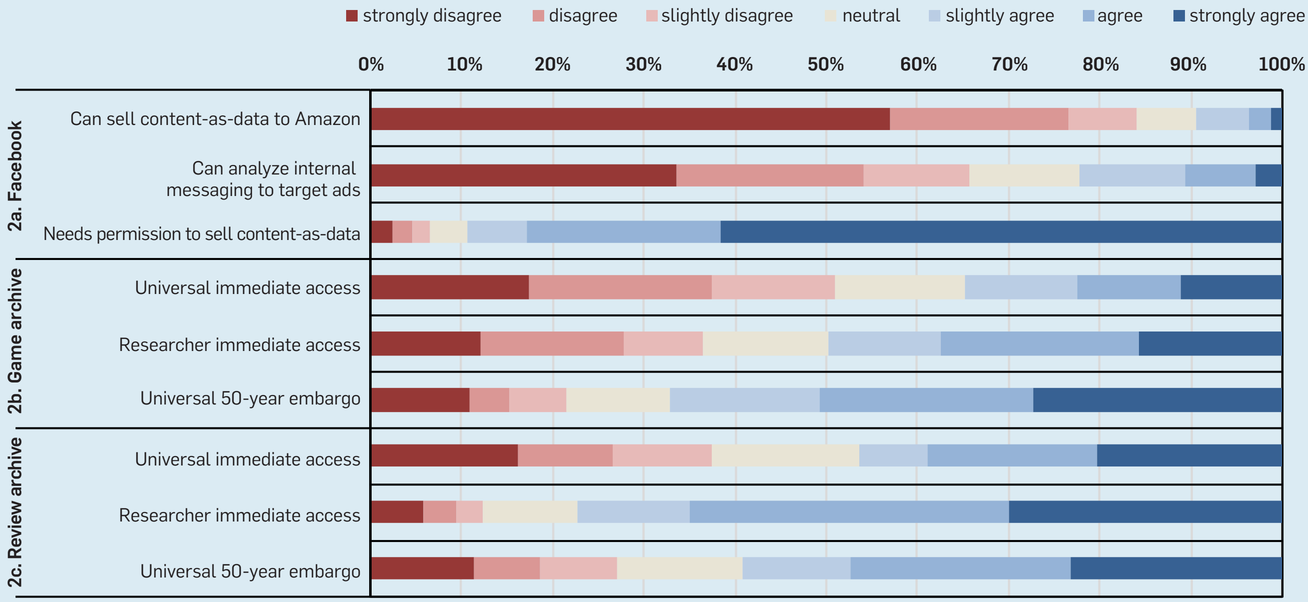
Humor is important when content is reused. Our study participants seemed to feel that relatively few ownership restrictions should be placed on lighthearted content that falls within prescribed social norms. On the other hand, reuse that is "mean," unwarranted, or offensive was judged more harshly. A "do no harm" heuristic provides a rough guide for how people propose to reuse other peoples' material.

Reuse for social good is viewed more skeptically than reuse of nominally humorous content. In each of our studies, a final scenario explored the idea of the U.S. Library of Congress acquiring public content from a type-specific social media service (such as Amazon Book Reviews, Facebook, Flickr, and YouTube) to create an historical collection. Three or more associated hypotheticals included in each survey tested different access restrictions on these archives. Results showed participants were more satisfied if collection access was restricted to researchers or embargoed for a 50-year period. Our own further examination of results revealed participants wanted to maintain long-term control of their public content.

Study participants are increasingly aware of what they give up when they publish profiles that describe themselves. In addition to preventing privacy

Figure 2. Social norms for content reuse.

2a shows three social network hypotheticals that test commercial reuse of personal content as data. 2b and 2c compare two archival collection development scenarios, both assumed to be socially beneficial. 2b shows participant responses to proposed access limits on a collection of gaming data, And 2c shows the same limits applied to a collection of logs of gameplay.



loss, participants also seek control over their digital identities. An important reason participants do not want the Library of Congress to maintain social media collections is that their old, unrevised, public selves may be on display in such collections; ordinary citizens wish to control the retrospective and future versions, not just the current version, of themselves. Media types more closely associated with one's sense of self—Facebook profiles, photos, tweets, gaming data, even reviews—provoke a stronger anti-collection response. For media types more aligned with personal privacy—photos, tweets, gaming data—survey participants reported that limiting access to researchers—even without a definition of what constitutes a researcher—mitigates anticipated harm; for media types strongly aligned with one's identity, the preferred mitigation strategy is to place the collection under a long-term embargo restricting access to the collection for a specified period. Figure 2b and Figure 2c compare reactions to retaining a collection of gaming data as opposed to a collection of online book reviews. Gaming data provokes a strong anti-collection response if everyone is given immediate access, mitigated somewhat by limiting access to researchers, and even more by putting the collection under a 50-year embargo. On the other hand, the harm associated with collecting online book reviews is better mitigated by limiting access to the collection to researchers, and less by the proposed 50-year embargo.

This finding returns the discussion to the issue of intent; to garner popular support, reuse for the public good must be weighed against individuals' ability to develop and maintain a sense of digital identity. Likewise, notions like veracity, which normally do not enter into the legal calculus of fair use, do play a part in defining social norms.

Figure 2 summarizes important reuse concepts, as they give rise to social norms by providing contrasting responses to pairs or triples of hypotheticals.

Removing Social Media

Removing social media content by anyone besides the person who posted it is the most speculative of the three actions we have investigated. Our surveys refer to this action as "removal" rather than "deletion" because it is intended to be nondestructive. Removal targets the copy in a particular place or context, not the content itself. Through their responses to open-ended questions about content removal, participants' revealed they usually remove material for three curatorial reasons: as a personal information management task (such as "cleaning up" one's account); in service of online identity management (such as untagging an unflattering photo of oneself); or to reflect one's changing understanding of privacy or some other aspect of online life (such as removing one's birthday from a profile).

Most social media services do not allow users to remove content created or posted by someone else. We thus derived hypotheticals from what participants in other studies mentioned they wanted to do.[7,19] Our surveys' removal hypotheticals primarily tested three variants: changes in the remover's relationship to the content (such as Should social media users be able to remove published content according to their own self interest?); circumstances in which a neutral non-owner can remove material (such as Should non-owners be able to remove published content they believe is demonstrably wrong); and situations in which removal requires requesting and receiving permission.

Participants generally did not support the idea of removing someone else's content in one's own self-interest, regardless of whether other mitigating circumstances were introduced.

What about the case of fraud detected by a neutral non-owner? Wikipedia has inured its users to the idea that content will be reviewed and removed if it does not pass the acid test. The results of our studies have revealed that certain entities are imbued with sufficient authority to support this hypothetical type of removal. As a social norm, veracity is apparently balanced with the first factor—self-interest. If self-interest is involved, a content non-owner is not entrusted with the public welfare.

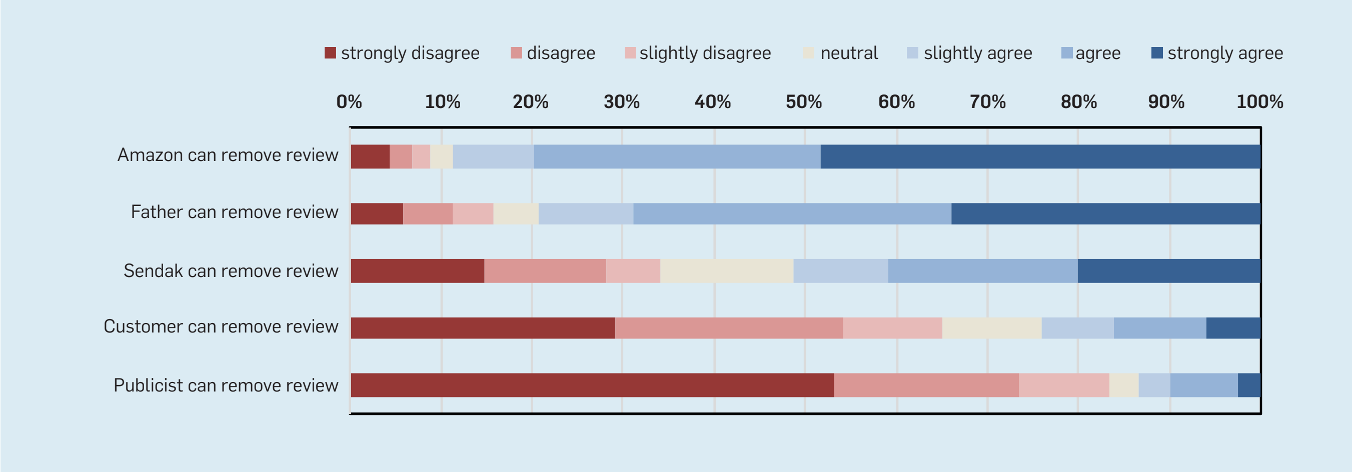
Our study participants have often felt a custodial relationship to the content (such as a website owner, service provider, or podcast producer) should give users the authority to remove content. In fact, the common expectation is that commercial service providers should not only have the authority to remove false or inaccurate content, they should also be charged with the responsibility for doing so.

A book-review scenario provides good examples of this test. In it, a seven-year-old girl has posted a negative book review that turns out to have been written by her father. Who has the authority to remove it? Hypotheticals we spelled out in our surveys tested removal by six different entities: the commercial host of the review website (Amazon); two potential content owners (the father or the girl); the book's author (Maurice Sendak); Sendak's publicist; and an Amazon customer who learned the review was posted under false circumstances. Who do participants believe should be given prevailing authority to remove the content, given a good reason to do so?

Study participants endorsed the idea of granting Amazon, the commercial host of the reviews, the responsibility of removing dubious content (89%, or 180/203, positive responses). The father or the girl (now a teen), as content owners, were secondarily given the authority to remove the review (77% and 79% positive, respectively). In spite of the question of self-interest, 51% of the participants thought Sendak (the author) should be able to remove the apparently fraudulent review. Only 24% thought a knowledgeable customer (aware of the circumstances under which the review was written) should normally be allowed to remove the review. Unsurprisingly, the least popular option was to allow the publicist, who was clearly acting in her client's self-interest, to remove the review; 83% thought this should not be allowed in normal circumstances. Figure 3 compares the action taken by different stakeholders.

Removal is generally the most controversial of the three actions. Although social media users want to be able to groom their own online self-presentation, there is a concomitant expectation that others should not remove content willy-nilly. The surprise in the study participants' responses to these hypotheticals was the degree of authority invested in the commercial service providers. The norm is to see commercial service providers as content guardians when such an action

Figure 3. Content-removal norms comparing five potential actors. Hypotheticals refer to a scenario in which an unfavorable book review submitted by a child was actually written by the child's father.



is necessary. Study participants also expressed intolerance for content removal motivated by self-interest.

Conclusion

We designed these eight studies, conducted over the course of five years, to elicit attitudes about how user-contributed content may be saved, reused, and removed by people other than the content's most obvious owner. The results reveal how far public attitudes have strayed from conventional legal concepts and how much they are tied to media type and other circumstantial factors. Yet these attitudes are surprisingly robust, regular, and predictable, suggesting emerging norms for the ownership and control of social media.

Among the highlights of our findings, which can be media-type-specific, are five recurring social norms:

Save anything but respect explicit social constraints. Study participants have felt they have the right to save almost anything they encounter on the open web. They reject notions of artificially imposed limits on the right to save content but also respect the explicit constraints introduced by a social network like Facebook. Social distance imposes a strong effect on whether a person can save personal information posted on Facebook. Even the powers of a friend of a friend are limited. Ownership effects are thus stronger inside social networking services than they are outside, on the open Web;

Concern for control of self-presentation. Participants object to proposals for the institutional archiving of public content on social media services like Twitter and Facebook. When probed, their objections stem less from conventional notions of privacy loss than from the loss of control of their own digital self-presentation. Digital self-presentation is subject to ongoing revision through deletion and curation of relevant online material.[7] Congruent with Samuelson's analysis of personal data as intellectual property,[18] our study participants have felt a strong right to own and control their own digital footprints, regardless of the source of this content;

Reuse norms reflect some aspects of fair use and ignore others. Social norms for reuse reflect many of the nuanced concerns of fair use (such as limiting commercial reuse, and encouraging creative or educational use) but are more pragmatic and often more conservative, relying excessively on the mitigating effects of permission. Original intent and original context are often part of a calculus of circumstantial fairness;

A right to veracity. Our study participants take information veracity seriously, though they put excessive responsibility for it into the hands of infrastructure providers. Content removed in blatant self-interest falls under this rubric, and participants generally deny others the right to remove content if self-interest is the only rationale. Fairness and accuracy are, however, seen by the participants as part of a right to remove content; and

Highly circumstantial reasoning about reuse. Differences among responses to the varying hypotheticals demonstrate that participants' sense of media rights may be highly dependent on the actual reuse situation. This contextual sensitivity may interact with labeling systems like Creative Commons, since people may be unable to conceive of the full range of possible reuse scenarios or predict which are most probable. As mentioned earlier, "permission" is some participants' go-to way of mitigating unpredictable reuse. Not only does fair use case law make such a workaround unnecessary, it also does not scale to viral reuse, and experience suggests permission from the original content creator is often unobtainable.[14]

For example, Etsy artists may explicitly permit noncommercial use of their work, since they envision reuse that promotes their art. In so doing, they may fail to consider a popular crafts parody site that pokes fun at artisanal work. Although buyers flock to the artists' stores as a result of their work's exposure to a new audience (in line with their intent), artists may still feel indignant about the nature of the reuse. To complicate matters further, the parody site donates its proceeds to charity, so the sting is mixed with social good. Content creators are thus faced with complex trade-offs. Whether reuse restrictions are implemented through technology, policy, or a combination of the two, managing rights relies crucially on the ability of content owners to envision plausible reuse scenarios and predict which are most likely.

In *Code: Version 2.0*, Lessig[6] identified four constraints that regulate online behavior: architecture, law, market forces, and social norms. As seen in the overall responses in our studies, social norms seem to have an outsized effect on participants' perceptions of what they (and others) can do with user-contributed content. Even social-media-savvy participants have little understanding of the relevant legal guidelines. Software-based governance is easy to ignore or thwart. And much reuse is oblivious to market forces. Furthermore, social norms are often nonreciprocal in action; participants in our studies did not always apply the same standards to themselves that they did to others, especially in non-abstract practical situations. This lack of reciprocity is not uncommon in other aspects of online behavior and may be attributed to individual users' ability to reflect on their own motives and intentions but not those of others.

What are the design and policy implications of these results? For one, they signal certain design gaps when media creators use labeling schemes (such as Creative Commons[2]); study participants seemed more sensitive to actions like reuse when they are offered examples rather than abstract labels. Hypothetical examples of reuse, especially those based on the media being labeled, may be helpful for extending Web users' understanding of the abstract ideas expressed by labels. It is no accident that our final norm addresses highly circumstantial factors as the nature of the content (such as "Is it personal?"), the differential scope of the audience (such as "Is the content going viral or is it playing to an audience of 10?" and "How different is the scope from the original?"), the type of reuse (such as is the content used in a way that highlights the original intent?), and the way the implied (or explicit) social contract between all potential owners of both the original and derived work is handled (such as "Is attribution or anonymity desired?").

Note only one of these factors—the nature of the content—is known at publication time, or the time when content is usually labeled. Other factors depend on how the content is reused (such as changes in genre, audience, or publication venue). Still others are not revealed until time has passed (such as the differential scope of the audience). That these factors are crucial to how a labeling scheme is used makes us think that supplemental mechanisms might be desirable; scenarios, hypotheticals, and mixed-initiative dialogs help content creators better envision many types of reuse or decide between attribution or anonymity or triggers that reveal when the scope or audience has changed. Still others depend on, say, the motivations for storing content. Past work tells us that individuals archive work that is not their own just as surely as institutions do.[15]

Ownership-driven questions need to be approached thoughtfully, lest we impose legal restrictions when none are necessary or fail to anticipate normal actions that will trigger reactions that could have been averted. Gaps between desired policy and current social norms may yet be bridged through education and thoughtful design.

Acknowledgments

We would like to thank Microsoft Research, Silicon Valley, for supporting the studies on which this article is based. Thanks, too, to the study participants for their patient and thoughtful completion of the lengthy surveys.

References

1. Aufderheide, P, Jaszi, P., Bieze, K., and Boyle, J.L. *Copyright, Free Speech, and the Public's Right to Know: How Journalists Think About Fair Use.* SSRN, Elsevier, July 30, 2012; http://ssrn.com/abstract=2119933
2. Boyle, J. *The Public Domain: Enclosing the Commons of the Mind.* Yale University Press, New Haven, CT, 2008.
3. Ellickson, R. *Order Without Law.* Harvard University Press, Cambridge, MA, 1994.
4. Fiesler, C. and Bruckman, A.S. Remixers' understandings of fair use online. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing* (Vancouver, BC, Canada, Mar. 14–18). ACM Press, New York, 2014.
5. Hill, B., Monroy-Hernandez, A., and Olson, K. Responses to remixing on a social media sharing website. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media* (Washington, D.C., May 23-26). AAAI, 2010, 74–81.
6. Lessig, L. *Code: Version 2.0.* Basic Books, New York, 2006.
7. Lindley, S., Marshall, C.C., Banks, R., Sellen, A. and Regan, T. Rethinking the web as a personal archive. In *Proceedings of the 22nd International World Wide Web Conference* (Rio de Janeiro, Brazil, May 13–17). ACM Press, New York, 2013, 749–760.
8. MacCormick, D.N. and Summers, R., Eds. *Interpreting Precedents.* Ashgate/Dartmouth, Farnham, U.K., 1997, 528–9.
9. Marshall, C.C. and Shipman, F.M. Social media ownership: Using Twitter as a window onto current attitudes and beliefs. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada, May 7–12). ACM Press, New York, 2011, 1081–1090.
10. Marshall, C.C. and Shipman, F.M. The ownership and reuse of visual media. In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries* (Ottawa, Canada, June 13–17). ACM Press, New York, 2011, 157–166.
11. Marshall, C.C. and Shipman, F.M. Experiences surveying the crowd: Reflections on methods, participation, and reliability. In *Proceedings of the Fifth Annual ACM Web Science Conference* (Paris, France, May 2–4). ACM Press, New York, 2013, 234–243.
12. Marshall, C.C. and Shipman, F.M. Saving, reusing, and remixing web video: Using attitudes and practices to reveal social norms. In *Proceedings of the 22nd International World Wide Web Conference* (Rio de Janeiro, Brazil, May 13–17). ACM Press, New York, 2013, 885–896.
13. Marshall, C.C. and Shipman, F.M. Exploring the ownership and persistent value of Facebook content. In *Proceedings of the 18th ACM Conference on Computer-Supported Cooperative Work and Social Computing* (Vancouver, BC, Canada, Mar. 14–18). ACM Press, New York, 2015, 712–723.
14. McDonough, J., Olendorf, R., Kirschenbaum, M., Kraus, K., Reside, D., Donahue, R., Phelps, A., Egert, C., Lowood, H., and Rojo, S. *Preserving Virtual Worlds Final Report. National Digital Information Infrastructure and Preservation Program, Washington, D.C.*, Aug. 31, 2010; http://hdl.handle.net/2142/17097
15. Odom, W. Sellen, A., Harper, R., and Thereska, E. Lost in translation: Understanding the possession of digital things in the cloud. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Austin, TX, May 5–10). ACM Press, New York, 2012, 781–790.
16. Rissland, E.L. Dimension-based analysis of hypotheticals from Supreme Court oral argument. In *Proceedings of the Second International Conference on AI and Law* (Vancouver, BC, Canada, June). ACM Press, New York, 1989, 111–120.
17. Sag, M. Predicting fair use. *Ohio State Law Journal 73*, 1 (2012), 47–91.
18. Samuelson, P. Privacy as intellectual property? *Stanford Law Review 52*, 1125 (1999).
19. Sas, C. and Whittaker, S. Design for forgetting: disposing of digital possessions after a breakup. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Paris, France, Apr. 27–May 2). ACM Press, New York, 2013, 1823–1832.
20. Shipman, F.M. and Marshall, C.C. Are user-contributed reviews community property? Exploring the beliefs and practices of reviewers. In *Proceedings of the Fifth Annual ACM Web Science Conference* (Paris, France, May 2–4). ACM Press, New York, 2013, 386–395.
21. Shipman, F.M. and Marshall, C.C. Creating and sharing records of multiplayer online game play: Practices and attitudes. In *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media* (Ann Arbor, MI, June). AAAI Press, Palo Alto, CA, 2014, 456–465.
22. Tehranian, J. Infringement nation: Copyright reform and the law/norm gap. *Utah Law Review 3* (2007), 537–550.

Catherine C. Marshall (ccmarshall@cse.tamu.edu) is an adjunct professor of computer science and engineering and affiliate of the Center for the Study of Digital Libraries at Texas A&M University, College Station, TX; she lives in San Francisco, CA, and volunteers at the Internet Archive (https://www.archive.org/).

Frank M. Shipman (shipman@tamu.edu) is a professor of computer science and engineering and associate director of the Center for the Study of Digital Libraries at Texas A&M University, College Station, TX.

Watch the authors discuss their work in this exclusive *Communications* video.
http://cacm.acm.org/videos/who-owns-the-social-web

DOI:10.1145/3064940

RRI requires doing the best science for the world, not only the best science in the world.

BY MARINA JIROTKA, BARBARA GRIMPE, BERND STAHL, GRACE EDEN, AND MARK HARTSWOOD

Responsible Research and Innovation in the Digital Age

AT A TIME when increasingly potent technologies are being developed with the potential to transform society, researchers in all technological fields, including information and communications technology (ICT), are under growing pressure to consider and reflect on the motivations, purposes, and possible consequences associated with their research. This pressure comes from the general public, civil society, and government institutions. In parallel is a growing recognition that current ethics review procedures within ICT may not address broader concerns (such as the potential societal consequences of innovation).

Instances of ICT raising concerns abound. For example, along with attention-grabbing headlines that artificial intelligence (AI) could ultimately pose an existential threat to humankind, there are more prosaic, yet strongly felt, social transformations already associated with AI technologies. For example, AI is an increasingly powerful protagonist in the story of how digital technologies are transforming the nature of work, as more types of work are mediated digitally, including how it is allocated, assessed, and rewarded. With these new forms of digital agency driving important aspects of labor markets, questions arise as to whose interests are being served and how accountability and transparency can be assured.

This is but one example of many debates around technology-, product-, and process-based innovation. Potential conflicts are wide-ranging and, most important, often emerge only after technologies have been embedded into the mainstream.

ICT scholars and professionals have long tried to understand and address these issues, though there are still numerous areas of concern. A novel concept—"responsible research and innovation," or RRI —has emerged recently in response to the challenge of designing innovations in a socially desirable and acceptable way. It may be useful for framing the discussion about how to manage the introduction of future innovations in ICT. In this article, we discuss the origins of RRI, consider relevant research from computer ethics and human-computer interaction (HCI), and illustrate the need for a new approach for the governance of ICT research. Finally, we suggest ways researchers might draw upon a framework for RRI in ICT based on the findings of an interview study conducted

» key insights

- ■ Responsible research and innovation aims to ensure that the processes and outcomes of research are aligned with societal values.
- ■ Our qualitative interview study found that ICT raises particular challenges for RRI; ICT researchers are best positioned to be able to identify them.
- ■ We propose a context-specific and flexible framework for RRI in ICT to help researchers use it as a source of inspiration and creativity.

IMAGE BY SUAT GURSOZLU

with the ICT community by investigators at the University of Oxford and De Montfort University, both in the U.K., from 2011 to 2013.

Ethics and Social Responsibility

ICT has traditionally been associated with the development of tools with discrete and transparent functionality aimed at supporting specific tasks. However, its diversity, scope, and complexity have extended far beyond this view to become situated within the very fabric of our daily lives.[17] Rather than being merely tools, the technologies now being designed are arguably transforming and augmenting the world around us, where computer-generated information, objects, and infrastructures "coexist in the same space as the real world," as outlined by Azuma et al.[1]

Debates about ethical issues in ICT are not new; researchers have been concerned with ethics in computing since at least the 1950s.[23] With the emergence of HCI in the 1980s, these debates have focused on the design of usable interactions between people and computers, where the broader ethical and societal aspects of application design and use have also been considered.[4] ICT researchers have tried to address ethical questions in many ways, as in, say, participatory design[13] and ICT for development.[10]

In addition to the approaches to ethics that come from within the ICT research and development community, there is a rich array of complementary thought that likewise tries to address particular ethical issues. The field of computer ethics, which draws on philosophy and social sciences, as well as computer science and information systems, has a history of reflecting on the ethics of ICT.[6,11]

Professional bodies like ACM (https://ethics.acm.org/), IEEE (http://www.ieee.org/about/ethics.html), and BCS (http://www.bcs.org/category/6030) have developed codes and standards for professionals to follow when considering ethical issues. While guidelines and standards are in place, there is an ongoing debate in regards to the limits of these approaches. A key question is whether or not future ethical and societal challenges are likely to be amenable to being addressed this way.

Though these approaches to identifying and debating ethical conflicts and questions are valuable, what is lacking today is a way to combine them in a manner that will allow a broad range of stakeholders to systematically engage with the goals, purposes, challenges, problems, and solutions encountered in research and innovation processes. This means individual researchers, research institutions, professional bodies, research funders, industry, and civil society all need to collaborate more. In practice, it means incorporating different kinds of knowledge, including from citizens, to inform the goals, directions, and trajectories of innovation in an inclusive way. This has been the case in some areas, as in, say, privacy and data protection, where longstanding debates have led to regulation and legislation, and to innovation in methods for design. However, such processes of collective reflection and deliberation have not yet happened in many areas of ICT. In light of the societal importance of ICT, broader engagement may now be necessary. Other areas of research and innovation that have been more socially contested have a longer history of engagement. We thus propose to look at RRI as a discourse that has evolved from these more contested fields, including whether and how it may be applied to ICT.

Scope of RRI

RRI initiatives across policy, academic research and scholarship, and legislation emerged more than a decade ago.[5,15] It began by aiming to identify and address uncertainties and risks associated with novel areas of research, beginning with nanotechnology[5] and moving to the environmental and health sciences, including geo-engineering[18] and synthetic biology.[21] The scope of RRI has since expanded to include computer science, robotics, informatics, and ICT more generally.[8] RRI proposes a new process for the governance of research and innovation. The aim is to ensure science and innovation are undertaken in the public interest by incorporating methods for encouraging more inclusive and democratic decision making through greater inclusion of stakeholder communities that might be directly affected by the introduction of novel technologies.

That is, RRI proposes a more reflective and inclusive research and innovation process, from fundamental research through to application design. In each phase of the innovation process, certain responsibilities may be associated with activities that occur within them, particularly in relation to how decisions taken might affect society. The focus is on creating a new mode of practical research governance that would transform existing processes, ensuring greater acceptability and even desirability of novel research and innovation outcomes, while also identifying and managing potential risks and uncertainties. RRI requires widening the scope of research and development from governance of risk to governance of innovation itself.[18]

There is a broad debate over the conceptual foundations of RRI and ways to implement it in practice. The most advanced framework for RRI today is probably the one proposed by Stilgoe et al.,[18] who also provided a non-exhaustive list of possible RRI methods, tools, and techniques (such as citizen juries and moratoriums). This approach has been taken up in E.U. policy and research, as in the RRI Tools project (https://www.rri-tools.eu/). It has also been adopted and adapted by the U.K. Engineering and Physical Science Research Council (EPSRC, https://www.epsrc.ac.uk/research/framework/). The EPSRC framework uses the acronym AREA to describe four key RRI components: Anticipate possible outcomes of research and innovation, Reflect on motivations, processes and products, Engage with relevant stakeholders, and Act accordingly to address issues revealed.

The ideas behind RRI and the AREA framework may be easy enough to understand but raise significant conceptual and practical questions. Fundamental problems include the fact that research and innovation do not follow linear and predictable patterns. Bunching together research and innovation blurs important boundaries and hides significant differences. To complicate matters, pluralistic democracies usually lack consensus as to what counts as acceptable and desirable. Additionally, stakeholder engagement can be misused for specific aims. The idea of RRI itself contains specific values, and implementing it may engender power struggles.

Most participants in the RRI discourse are well aware of these issues.[14] It is thus important to understand that RRI is not an attempt to invent a new top-down way of governing research and innovation but rather a way of linking and embedding existing principles and activities with a view to broadening their reach and relevance. This means RRI encompasses existing techniques for public engagement and reflection (such as participatory design, research ethics, and professional codes) and aims to ensure they can develop synergies. It also means building on extant research into corporate ICT governance. More precisely, RRI may be understood as a demand for multilevel ethics (systemic and institutional macro ethics, in addition to individualistic micro ethics), engagement of a broader variety of stakeholders, and inclusion of social, political, and ethical issues in ICT governance.[7] It remains problematic, though, how these ideas can be put into practice.

Embedding RRI in ICT Innovation

Embedding RRI into ICT innovation is a challenge. First, it is necessary to understand how ICT researchers and practitioners manage their professional responsibilities, as well as how they perceive the notion of RRI, in order to assess how to move forward and fit features of RRI to researchers' perceptions and expectations. One significant issue is how to develop a set of practical actions within an RRI framework that may be adopted by the ICT community and how it might be embedded and deployed within current organizational processes. In order to address these questions, we conducted investigations from 2011 to 2013 with ICT researchers in the U.K. among research funders, professional organizations, industry, and civil society organizations into the ways RRI concepts, tools, and processes might become a creative resource for innovation in ICT. Our work was part of the Framework for Responsible Research and Innovation in ICT project funded by the Engineering and Physical Sciences Research Council (EPSRC, https://www.epsrc.ac.uk/) in the U.K.

RRI proposes a more reflective and inclusive research and innovation process, from fundamental research through to application design.

The ICT Community Landscape

We interviewed leading computer scientists, researchers, and postdoctoral and Ph.D. students, as well as EPSRC portfolio managers and representatives of professional bodies in the U.K.[3] The study was the first extensive summary of current positions regarding the boundaries of professional responsibility and identification of potential long-term societal consequences of ICT. It provides an important baseline, giving us an opportunity to describe, understand, and triangulate ICT researchers' and other stakeholders' questions and concerns across a variety of computer science domains, including mobile computing, AI, photonics, and signal processing.

Many researchers welcome enhancements to current governance processes (such as by framing questions that help reflect on research outputs). Also, some researchers embrace the further integration of social and ethical research into design and development. Apart from such perceived RRI opportunities, many interviewees in our study raised concerns. We outline five, as discussed by participants. Together, they identify typical problems involved in integrating RRI into ICT. We thus sought to relate them to concepts and approaches that would allow researchers to specify RRI in ICT.

The first is the difficulty of predicting the potential uses of research outcomes. Some researchers we interviewed said it may be inappropriate to attempt to predict future effects in the context of ICT research because the uncertainties tend to be social rather than scientific, meaning technologies are socially shaped and not fixed. Researchers in the study cited two unknown factors related to prediction. First, in fundamental research, risks and uncertainties are identifiable only within the context of their use. Second, in application-oriented research, industry and user adaptation can change the trajectory of ICT in unforeseen ways. The very open nature of ICT, its logical malleability,[12] interpretive flexibility,[2] and the social production of technology make it even more difficult to predict outcomes of research and innovation than in other areas of science and technology research. We refer to these aspects of ICT as related to the "product" of ICT research and innovation.

A second concern points to the perceived differences between ascertain-ing risks and uncertainties in computer science to that in the physical and life sciences. For example, researchers participating in our study discussed what we refer to as the "rhythm of ICT" whereby outputs may occur at a quicker pace than in the physical sciences. Software may be developed, released, and go viral potentially on the same day with little, if any, oversight and have far-reaching effects on human activities and societal structures. These concerns relate to the "process" of research and innovation.

A further distinguishing feature typical of ICT is what Johnson[11] called "the problem of many hands," or organizational and institutional reliance on a division of labor whereby most activities are split among numerous individuals. The problem becomes more fraught beyond organizational boundaries when trying to conduct open source projects. Moreover, different disciplinary languages are significant, making interdisciplinary work that much more important but difficult to achieve in practice. Ascribing accountability for eventual consequences is therefore difficult. These aspects of ICT projects point to the importance of considering what we call the "people dimension" of RRI in ICT.

A final concern that emerged from our study is the notion of "convergence"[9] whereby the increasingly pervasive nature of technologies in the age of the Internet, Web 2.0, and pervasive computing means that demarcating clear boundaries among systems, features, and functionality is increasingly problematic. Blurring boundaries means it becomes progressively more difficult to discern the "purpose" of ICT research and innovation.

These concerns pose a significant challenge to RRI in ICT that may go beyond those in other fields. We thus developed the "4 Ps," or product, process, people, and purpose, outlined earlier, as well as other concepts and approaches to be explained next, to develop a framework for RRI specific to ICT.

Fundamental problems include the fact that research and innovation do not follow linear and predictable patterns.

Toward AREA Plus

The AREA acronym refers to general points of interest in RRI, but more detail is needed for ICT research. The discussion so far has shown that RRI in ICT cannot be realized in a prescriptive manner. The nuances of acceptability and desirability and competing interests and their embedding in social, economic, and political structures mean that many aspects of ICT are likely to remain contested for the foreseeable future. RRI cannot therefore expect to establish universal definitions of what counts as responsible but instead needs to be understood as a contextual process that facilitates development of sensitivities toward relevant issues and a willingness of stakeholders to engage with one another, making them responsive to mutual needs and interests.

We frame RRI for ICT as an ongoing cultural dialogue in which multiple voices from within the HCI community talk to RRI proponents in order to find ways of translating back and forth what forms of responsible ICT design and development might already be available, be under development, or have yet to be developed. This approach is akin to the view asserted by Strand et al.[20] who developed a set of indicators for the European Commission that could be used to monitor RRI across different disciplines, research themes, and projects. While proposing a comprehensive list of indicators, Strand et al. also suggested that any indicator set would ultimately need to be (re)developed in a given research or application context. Our framework is thus self-critical by design and meant to be continuously challenged and adjusted.

We exemplify what such a dynamic and context-sensitive framework for responsible behavior might include for ICT. Our EPSRC-funded study focused on interviewees' comments regarding the difficulty of predicting ICT trajectories. While we regard this as appropriate skepticism in the overall RRI discourse, under "anticipation" of socio-technical futures we also suggest different approaches that consider the possible futures their innovation may bring about (such as a collaborative quest for future solutions informed by current experiences). This alternative view profits from existing ICT research; that is, ICT researchers have much to add to the RRI discourse to make it more context-specific and useful. Reeves's analysis[16] of "envisioning" techniques is a case in point, making

Figure 1. The AREA Plus framework.

	Process Rhythm of ICT	**Product** Logical malleability and interpretive flexibility	**Purpose** Convergence and pervasiveness	**People** Problem of many hands
Anticipate	Is the planned research methodology acceptable?	To what extent are we able to anticipate the final product, future uses, and impacts? Will the product be socially desirable? How sustainable are the outcomes?	Why should we pursue this research?	Have the right stakeholders been included?
Reflect	What mechanisms are used to reflect on process? How might we do it differently?	How do we know what the consequences might be? What might be the potential use? What do we not know? How can we ensure social desirability? How might we do it differently?	Is the research controversial? How might we do it differently?	Who is affected? How might we do it differently?
Engage	How can we engage a wide group of stakeholders?	What are the viewpoints of a wide group of stakeholders?	Is the research agenda acceptable?	Who prioritizes research? For whom is the research being done?
Act	How can your research structure become flexible? What training is required? What infrastructure is required?	What needs to be done to ensure social desirability? What training is required? What infrastructure is required?	How might we ensure the implied future is desirable? What training is required? What infrastructure is required?	Who matters? What training is required? What infrastructure is required?

Figure 2. Selecting anticipation.

	Process Rhythm of ICT	**Product** Logical malleability and interpretive flexibility
Anticipate	Is the planned research methodology acceptable?	To what extent are we able to anticipate the final product, future uses, and impacts? Will the product be socially desirable? How sustainable are the outcomes?

Figure 3. Unpacking anticipation.

To what extent are we able to anticipate the final product, future uses, and impact?
The future cannot be predicted with certainty, but there is room for exploring different possible pathways. Also, researchers and other stakeholders can build on existing formal and informal practices of anticipation in the ICT community.

Exploring different possible pathways

- Who might be the intended audience(s) of the envisioned product?
- What is the context the envisioned product is meant to address? And what is the context in which this anticipation process itself is taking place?
- What current issues does the anticipation process target or could target?
- What can we learn from earlier (historical) anticipation processes?
- In pursuing a particular vision, what pathways might we also be shutting down? And what endpoints and current issues might be excluded?

(Scaffolding questions adopted and adapted from Reeves[16])

Envisioning in ICT
As in Reeves,[16] although it is difficult to predict the trajectory of ICT innovations, including outcomes, future uses, and impacts, ICT is a domain in which vision, utopia, predictions, promises, and hype have been produced for decades. Much of it has been done rather unconsciously, thus shaping the trajectories of ICT in ways that shut down alternative paths. Implicit powers are also at play. Narratives, teleology, and technological determinism proliferate but are not sufficiently reflected.

clear that the social shaping of technologies is at the heart of computer science, not external to it, as suggested by some of the interviewees in our study. Visions, utopia, predictions, promises, and hype have been produced for decades concerning how socio-technical futures may unfold, though much of it has been done rather unconsciously, thus shaping the trajectories of ICT in ways that shut down alternative paths. There are thus implicit human and technological powers at play. Narratives, teleology, and technological determinism proliferate but are not sufficiently reflected.

In practical terms, our framework draws on such existing approaches to ICT development and provides a variety of scaffolding questions. Each aspect of the framework expands into deeper questions, suggesting literature, more detailed discussion, and problematization of a particular aspect of ICT innovation. For instance, after scanning the framework as a whole (see Figure 1) a researcher might want to consider to what extent the effects of ICT development may be anticipated (see Figure 2 and Figure 3). Various links between approaches provide questions for exploring different possible pathways, a more comprehensive line of reasoning, and references.

Our framework is meant to be adapted to the context in which researchers and other stakeholders find themselves. The idea is to productively "open up" not "close down" expert discourse.[19] At the same time, we do not question "closure" per se. Any design-and-development process requires taking countless decisions and translating them into software and hardware solutions at multiple points in time. However, closures may still leave room for diversity.[19]

In sum, certain forms of practical self-reflection and self-criticism exist in ICT research and could be cultivated further under the extended AREA Plus framework. In this sense, EPSRC's original AREA principles are a starting point for the reinvigoration and possible extension of a much more nuanced discourse with and within ICT research.

Future AREA Plus Framework

The framework we started to develop in 2011, as explained earlier in this article, is not a panacea and cannot perform miracles. Many questions of relevance concerning ICT projects are related to fundamentally opposing concerns and socially and politically contested interests. Such conflicts will not disappear overnight. However, the framework may allow researchers and innovators to better understand their own and others' positions and contribute to better-informed debate and higher-quality policies and decisions.

Much remains to be done to achieve this vision of responsible technology development and support its progress. The framework needs to be supported by effective tools and specific guidance on particular topics, issues, and technologies. The web-based resource we developed to provide them (http://www.orbit-rri.org/) is only a starting point. We next identify concerns that are crucial to the further development and adoption of the framework.

First, embedding RRI activities needs to be perceived by researchers as something achievable. As we explained earlier, "anticipation" becomes significantly less mysterious when realistically scoped and grounded in concrete practices, including specific envisioning techniques and questions. Implementing RRI is about finding ways to instantiate concrete achievable practices and not about unattainable ideals of "perfect" foresight or "risk-free" innovation. Also, RRI for ICT may require developing new initiatives that are likely to depend on more fine-grain case studies beyond the scope of this article.

In addition, an integrated approach to RRI is needed for the successful adoption of the framework. RRI has to be sensitive to the relationships among researchers, practitioners, and the hierarchies and organizational structures in which they are situated. Responsibilities need to be apportioned across the entire ecology of organizations that together deliver research and innovation.[8] Taking RRI seriously as a strategic concern would facilitate practices of anticipation, reflection, and engagement to occur in the formation of new research programs by funding councils and in the final stages of commercialization at the academic/commercial interfaces where academic and commercial interests most visibly overlap and sometimes collide. In between these poles a responsible research and innovation process would incorporate the roles of funding councils, professional bodies, and others in sustaining RRI practices within research teams by providing appropriate support, services, and guidance. Responsible behavior thus becomes a collective, unpredictable activity, less about accountability and liability, and more about care and responsiveness to the public.[18]

There is evidence that these developments are under way. Academia and industry are starting to be aware of RRI for many reasons. Maybe the best of them, and a good conclusion for this article, is that RRI, while largely conceived as a risk-management approach to socio-technical change, has a much more positive trajectory than simply constraining innovation to mitigate risk. By incorporating active considerations of alternate socio-technical futures into design, engaging with stakeholders, reflecting on process, product, and purpose, and putting people at the center of research and innovation, RRI may well provide inspiration and become a unique source of innovation and creativity.

Acknowledgments

We would like to thank the reviewers for their thoughtful comments. The research underlying this article was funded by the EPSRC.

References

1. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. Recent advances in augmented reality. *IEEE Computer Graphics and Applications 21*, 6 (Nov. 2001), 34–47.
2. Doherty, N.F., Coombs, C.R., and Loan-Clarke, J. A re-conceptualization of the interpretive flexibility of information technologies: Redressing the balance between the social and the technical. *European Journal of Information Systems 15*, 6 (Dec. 2006), 569–582.
3. Eden, G., Jirotka, M., and Stahl, B. Responsible research and innovation: Critical reflection into the potential social consequences of ICT. In *Proceedings of the Seventh IEEE International Conference on Research Challenges in Information Science* (Paris, France, May 29–31). IEEE, 2013, 1–12.
4. Ehn, P. *Work-Oriented Design of Computer Artifacts.* Lawrence Erlbaum Associates, Mahwah, NJ, 1990.
5. Fisher, E. and Rip, A. Responsible innovation: Multi-level dynamics and soft intervention practices. In *Responsible Innovation*, R. Owen, J. Bessant, and M. Heintz, Eds. John Wiley & Sons, Inc., New York, 2013, 165–183.
6. Floridi, L., Ed. *The Cambridge Handbook of Information and Computer Ethics.* Cambridge University Press, Cambridge, U.K., 2010.
7. Gotterbarn, D. ICT governance and what to do about the toothless tiger(s): Professional organizations and codes of ethics. *Australasian Journal of Information Systems 16*, 1 (Nov. 2009), 165–184.
8. Grimpe, B., Hartswood, M., and Jirotka, M. Towards a closer dialogue between policy and practice: Responsible design in HCI. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems* (Toronto, Canada, Apr. 26–May 1). ACM Press, New York, 2014, 2965–2974.
9. Grunwald, A. Converging technologies: Visions, increased contingencies of the conditio humana, and search for orientation. *Futures 39*, 4 (May 2007), 380–392.
10. Heeks, R. ICT4D 2.0: The next phase of applying ICT for international development. *Computer 41*, 6 (June 2008), 26–33.
11. Johnson, D.G. *Computer Ethics, Third Edition.* Prentice Hall, Upper Saddle River, NJ, 2012.
12. Moor, J. What is computer ethics? *Metaphilosophy 16*, 4 (Oct. 1985), 266–275.
13. Muller, M.J. and Kuhn, S. Participatory design. *Commun. ACM 36*, 6 (June 1993), 24–28.
14. Owen, R., Heintz, M., and Bessant, J., Eds. *Responsible Innovation.* John Wiley & Sons, Inc., New York, 2013.
15. Owen, R., Macnaghten, P., and Stilgoe, J. Responsible research and innovation: From science in society to science for society, with society. *Science and Public Policy 39*, 6 (Dec. 2012), 751–760.
16. Reeves, S. Envisioning ubiquitous computing. In *Proceedings of the 30th Annual ACM Conference on Human Factors in Computing Systems* (Austin, TX, May 5–10). ACM Press, New York, 2012, 1573–1582.
17. Sellen, A., Rogers, Y., Harper, R., and Rodden, T. Reflecting human values in the digital age. *Commun. ACM 52*, 3 (Mar. 2009), 58–66.
18. Stilgoe, J., Owen, R., and Macnaghten, P. Developing a framework for responsible innovation. *Research Policy 42*, 9 (Nov. 2013), 1568–1580.
19. Stirling, A. 'Opening up' and 'closing down': Power, participation, and pluralism in the social appraisal of technology. *Science, Technology & Human Values 33*, 2 (Mar. 2008), 262–294.
20. Strand, R., Spaapen, J., Bauer, M., Hogan, E., Revuelta, G., and Stagl. S. *Indicators for Promoting and Monitoring Responsible Research and Innovation.* Publications Office of the European Union, Brussels, Belgium, 2015; http://ec.europa.eu/research/swafs/pdf/pub_rri/rri_indicators_final_version.pdf
21. Tucker, J.B. and Zilinskas, R.A. The promises and perils of synthetic biology. *The New Atlantis 12* (Spring 2006), 25–45.
22. Van den Hoven, J. Value-sensitive design and responsible innovation. In *Responsible Innovation*, R. Owen, J. Bessant, and M. Heintz, Eds. John Wiley & Sons, Inc., New York, 2013, 75–83.
23. Wiener, N. *The Human Use of Human Beings.* Da Capo Press, Cambridge, MA, 1954.

Marina Jirotka (MarinaJirotka@cs.ox.ac.uk) is Professor of Human-Centered Computing in the Department of Computer Science and Associate Director of the Oxford e-Research Center at the University of Oxford, Oxford, U.K.

Barbara Grimpe (barbara.grimpe@aau.at) is a postdoc assistant in the Alpen-Adria Universität Klagenfurt, Klagenfurt am Wörthersee, Austria.

Bernd Stahl (bstahl@dmu.ac.uk) is Professor of Critical Research in Technology and Director of the Centre for Computing and Social Responsibility at De Montfort University, Leicester, U.K.

Grace Eden (Grace.Eden@hevs.ch) is a senior academic associate in the University of Applied Sciences Western Switzerland, HES-SO Valais-Wallis, Switzerland.

Mark Hartswood (Mark.Hartswood@cs.ox.ac.uk) is a research assistant in the Department of Computer Science, University of Oxford, Oxford, U.K.

DOI:10.1145/3012008

Exploring the basic game theory models of contests found in online services.

BY MILAN VOJNOVIĆ

Contest Theory

A VARIETY OF Internet online services are designed based on contests. A canonical example is crowdsourcing services, which solicit solutions to tasks by open calls to online communities. Here the tasks can be of different categories, such as art design, software development, data-science problems, and various challenges such as planetary-scale locating of objects.[12,28] These services operate under certain contest rules that include specifying a prize allocation mechanism, for example, awarding only a first-place prize or several position prizes. The prizes can be monetary, or in-kind rewards such as in terms of attention, status, or computing resources, for example, CPU, bandwidth, and storage. We refer to a contest as any situation in which agents invest irreversible and costly efforts toward winning a prize, which is allocated based on relative performance. We use the term "contest theory" in a broad sense to refer to a set of theories developed for the better understanding and informed design of contests.

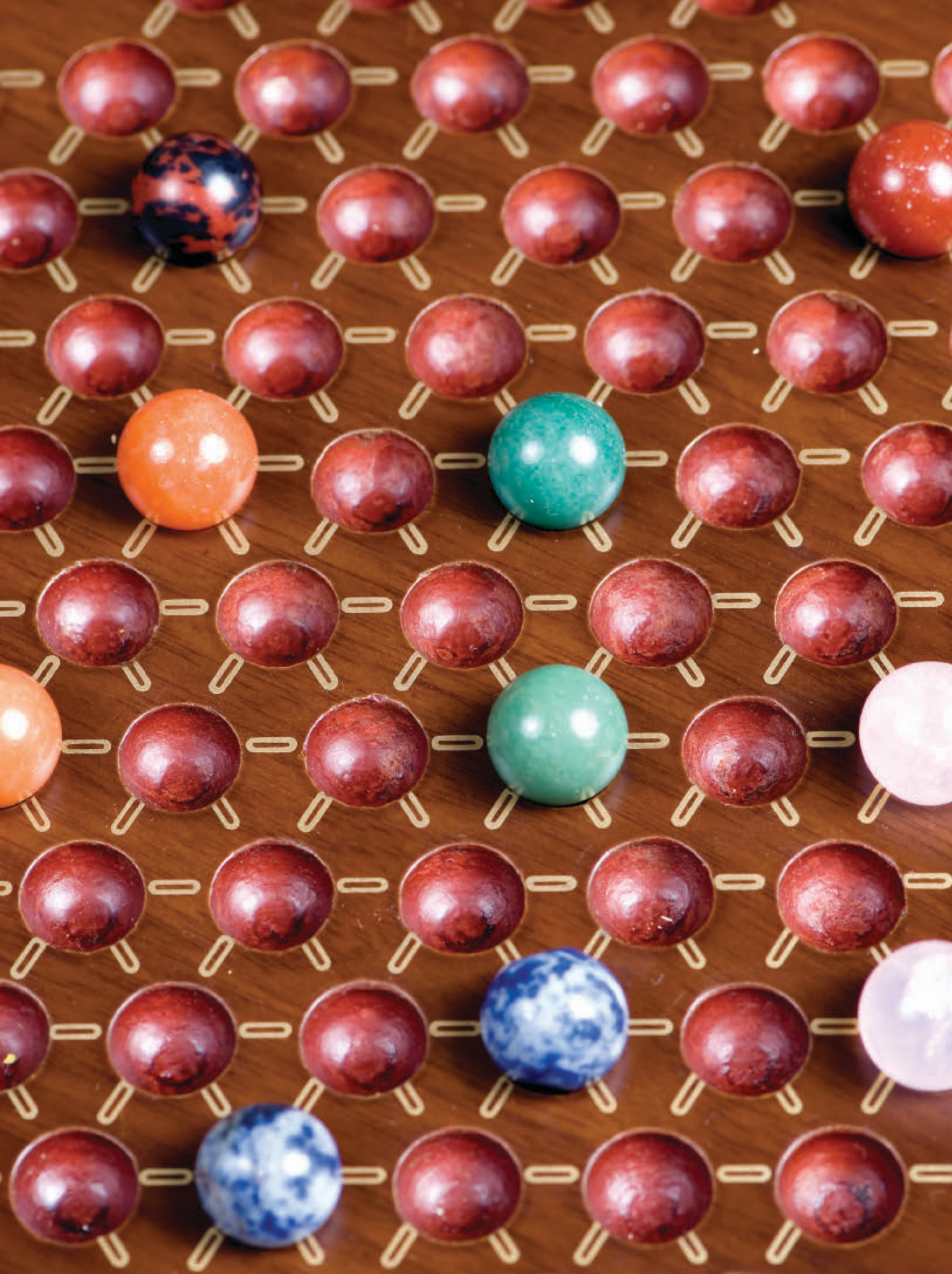
A central question in contest theory is: *How to allocate prizes to maximize a desired objective?* The objective may be to maximize the utility of production to the agent who solicits solutions to a task, or to the whole society. The question of how to allocate prizes was studied as early as 1902 by Galton.[15] A study of how to allocate prizes necessitates to consider the incentives of contestants, who act strategically in investing costly production efforts.[1,11,44] Game theory models of contests have been studied in auction theory, economic theory, operations research, as well as theoretical biology; for example, Bishop and Smith.[3] The use of compensation schemes based on an individual's ordinal rank rather than absolute performance in firms have been studied by economists; for example, Lazear and Rosen.[27] Game theory and pertinent computational questions have been studied by computer scientists.[31,33,36] Several new contributions have been made on optimal allocation of prizes in crowdsourcing contests, equilibrium outcomes in games that model simultaneous contests, and the worst-case efficiency of production in equilibrium outcomes of various games that model contests.

The skill-rating methods that use observations of relative performance comparisons as input, such as ranking outcomes in contests, have been studied extensively in the past. They are now widely used in various applications, such as sport competitions, online gaming, and online labor platforms.

» key insights

- **The operation of various online platforms relies on incentive mechanisms for eliciting user contributions, which take the form of a contest.**
- **Contest theory refers to a set of theories for the better understanding and informed design of contests.**
- **The theory provides insights into what user behavior may arise in equilibrium, guidelines of how to allocate prizes, and algorithms for estimating skills of individuals based on observed contest outcomes.**

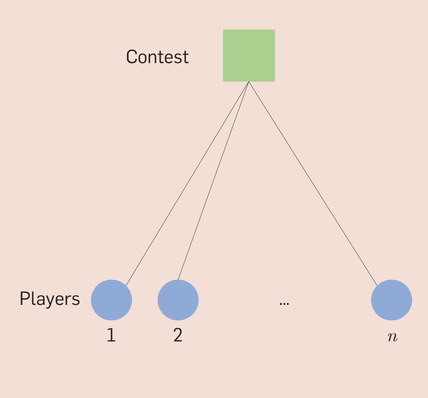
IMAGE BY THOMAS M. PERKINS



The design of skill-rating methods is based on statistical models of ranking outcomes developed from 1920s onward. More recent developments include skill-rating methods that allow for contests among two or more teams of players, which are common in online gaming and online labour platforms. New results have been recently developed in the area of statistical inference for statistical models of ranking data, including new characterizations of the accuracy of various skill parameter estimators and new iterative methods for skill parameter estimation.

In this article, we survey some main results of contest theory. Specifically, we discuss basic game theory models of contests that are found in online services. We explain the conditions under which to optimally allocate prizes to maximize a given objective, such as the total effort or the maximum individual effort, in a strategic equilibrium. We will focus on games in which players make simultaneous effort investments; the games that involve some aspect of sequential play are only briefly discussed.

Figure 1. Single contest.



We consider both games that model a single contest (see Figure 1) and games that model a system of two or more simultaneous contests (Figure 2). Simultaneous contests are common in the context of online crowdsourcing platforms. We explain basic principles of popular skill rating systems and point out some new results in this area. We conclude with an outlook on future research directions.

This article complements existing surveys on the game-theoretic aspects in contest theory, for example, Corchon,[9] Konrad,[25] and Nitzan.[32] We provide an overview of some of the topics covered in the book by Vojnović,[42] where the reader may find a more extensive coverage of references.

Strategic Game Models of Contests

The standard game theory framework for studying contests is based on the assumption that agents are rational and strategic players who invest effort with a selfish goal to maximize their individual payoffs. The payoff of a player combines the utility of winning a prize and the cost of production. Specifically, we consider a normal-form game that models a contest, defined by:

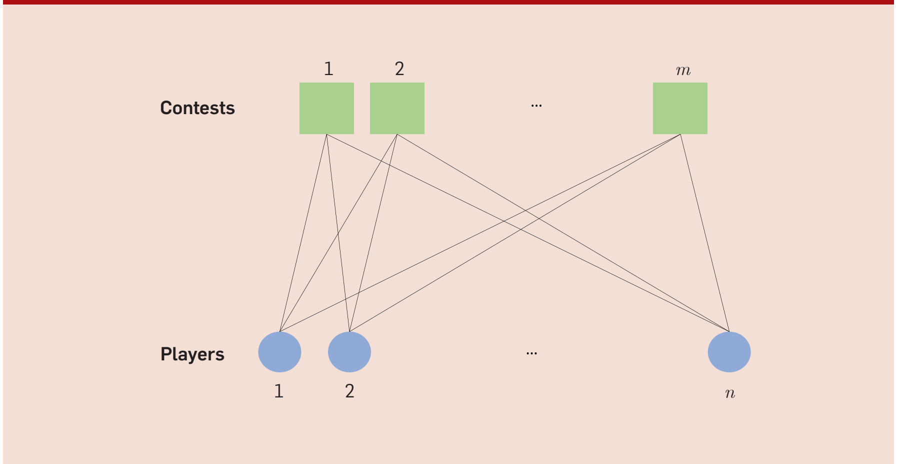
- *Set of two or more players:* $N=\{1,2,\ldots,n\}$;
- *Payoff functions:* for any given vector of efforts $\mathbf{b} = (b_1, b_2, \ldots, b_n)$, the payoff of player i is given by

  $s_i(\boldsymbol{b}) = v_i x_i(\mathbf{b}) - c(b_i)$

  where

  - $v_1, v_2, \ldots, v_n$ are positive-valued *skill parameters*,
  - $x(\boldsymbol{b}) := (x_1(\boldsymbol{b}), x_2(\boldsymbol{b}), \ldots, x_n(\boldsymbol{b}))$ is *prize allocation*, and
  - $c(x)$ is a *production cost function*.

Figure 2. A system of simultaneous contests: Edges indicate effort investment opportunities.



The skill parameters reflect the abilities of players: the larger the value of a player's skill parameter, the more proficient is the player. If the production cost is according to a linear function, we can normalize the payoff functions such that a player's skill parameter can be interpreted as the reciprocal of his or her production cost per unit effort. The game as defined here allows us to study equilibrium outcomes under different prize allocation mechanisms, such as assigning fixed shares of a prize budget in decreasing order of invested efforts, or splitting a prize budget among players in proportion to their effort investments. The prize allocation can be interpreted as the winning probabilities for an indivisible prize item, or as the shares of an infinitely divisible prize. The game allows us to study equilibrium outcomes for different types of production costs. For example, it is common to consider *linear production costs*, we refer to as *constant marginal production costs*, under which the production cost per unit effort is constant; in particular, we refer to *unit marginal production cost* when the production cost per unit effort is of unit value. We may also consider production costs with either *decreasing* or *increasing marginal costs*. It is noteworthy that the game as defined here formally corresponds to an auction, where efforts, skills, and production costs are in correspondence with bids, valuations, and payments, respectively

We say a game is *with complete information* if the players have perfect information about each other's skill parameters. A game with complete information can be used as a model of a contest when the players are informed about who is going to participate in the contest and about the skills of the participants. For example, a situation like this can be found in competition-based software development platforms such as TopCoder, where a contest takes place after a registration phase, which reveals identities of participants. A game is said to be *with incomplete information* if the value of each player's skill parameter is his or her private information. In a game with incomplete information, skill parameters are assumed to be random variables according to a prior distribution, which is a common knowledge.

A game with incomplete information allows us to model uncertainty about skills of competitors in a contest; in the context of online services, such an uncertainty may arise because it may not be a priori known who is going to participate in a contest.

The strategic effort investment by a player can be according to a pure strategy, specifying a value of the effort investment, or according to a mixed strategy, specifying a probability distribution over pure strategies. An investment of efforts by players is a *pure-strategy Nash equilibrium* if no player can increase his or her payoff by a unilateral deviation. Similarly, a set of mixed strategies is a *mixed-strategy Nash equilibrium* if no player can increase his or her expected payoff by a unilateral deviation. A *Bayes-Nash equilibrium* is a mapping of an individual's skill to a value of effort such that no player can increase his or her expected payoff by a unilateral deviation.

The utility of production is typically studied with respect to the following two metrics: the *total effort* and the *maximum individual effort*. The total effort has been studied extensively because it corresponds to the revenue accrued in an all-pay auction, and the total outlay accrued in a rent-seeking contest.[26,40] The maximum individual effort has been studied motivated by applications in contests, such as in crowdsourcing services, where a contest owner makes use only of the best submitted solution. The utility of production has also been studied from a societal perspective, defined by a *social welfare* function, which is commonly defined as the sum of payoffs of all the parties involved (players and the contest owner). For example, when players incur unit marginal production costs and the payoff to the contest owner is the total effort invested by the players, social welfare corresponds to the total valuation of prizes by those who win them. Social welfare in an equilibrium can be smaller than optimal value; in some instances, optimum social welfare is achieved only if a given prize budget is fully assigned to highest-skill players, while in equilibrium a lower-skill player can have a strictly positive winning probability.

A game with complete information can be used as a model of a contest when the players are informed about who is going to participate in the contest and about the skills of the participants.

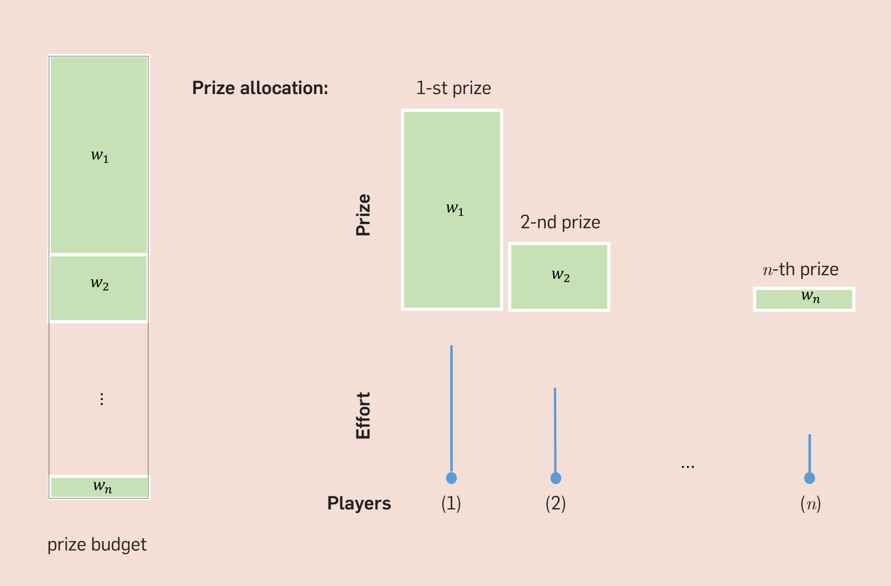
Single contest. We now consider a normal-form game that models a single contest among two or more players, for different prize allocation mechanisms and production cost functions. A model of a single contest allows us to study situations in which players have no outside options such as investing effort in an alternative contest; we will later discuss games that model simultaneous contests, which provide players with such outside options.

Standard all-pay contest. A classic game that models a contest, we refer to as the standard *all-pay contest*, assumes a prize allocation mechanism that allocates entire prize budget to a highest-effort player with random tie break, and unit marginal production costs. This game corresponds to the well-known game that models an all-pay auction, studied in auction theory. The given prize allocation mechanism is commonly referred to as *perfect discrimination*, because it assumes perfect identification of a highest-effort player, achieved by some flawless mechanism for comparison of individual efforts.

We first discuss Nash equilibrium outcomes in the game with complete information that models the standard all-pay contest. This game does not have a pure-strategy Nash equilibrium. It can be easily verified that for any given effort investments, there is always a player who has a beneficial unilateral deviation. On the other hand, the game always has one or more mixed-strategy Nash equilibria, which were first fully characterized by Baye, Kovenock, and de Vries.[2]

The game has a unique mixed-strategy Nash equilibrium only in some special cases, such as in a two-player contest, or in a contest with three or more players but where two players have individual skills larger than that of any other player. In general, the game has a continuum of mixed-strategy Nash equilibria. This may be considered a drawback because it implies a lack of predictive power. The mixed-strategy Nash equilibria are payoff equivalent: whenever a game has two or more mixed-strategy Nash equilibria, the expected payoffs in these equilibria are equivalent. In general, the equilibrium outcomes are not equivalent with respect to either the expected total effort or the expected maximum individual effort. It is noteworthy that

Figure 3. Rank order allocation of prizes: Allocation of fixed shares $w_1 \geq w_2 \geq \ldots \geq w_n \geq 0$ of a prize budget in decreasing order of effort.



there always exists a mixed-strategy Nash equilibrium in which all but two highest-skill players invest zero effort. The expected total effort in this equilibrium is at least as large as in any other equilibrium.

We now discuss some properties that hold in any mixed-strategy Nash equilibrium. Without loss of generality, assume that players' identities are in decreasing order of their skill parameters. The expected total effort is of value between $v_2/2$ and v_2. Interestingly, the expected maximum individual effort is always at least half of the expected total effort. This provides a theoretical support for the efficiency of competition-based crowdsourcing services in which a contest owner solicits solutions from multiple workers, but makes use only of the best submitted solution. Intuitively, one would expect that such a production system is bound to be highly inefficient because much of the invested work ends up being wasted. However, by this result, inefficiency can only be to a limited extent in any mixed-strategy Nash equilibrium. With regard to social welfare, there can be some efficiency loss in equilibrium, because a player whose skill is not the highest may have a strictly positive winning probability. However, this can only be up to a limited extent in any mixed-strategy Nash equilibrium: the expected social welfare is always at least 4/5 of the optimum social welfare.

Another noteworthy property is the so-called *exclusion principle*, which refers to the existence of game instances for which the expected total effort in equilibrium can be increased by excluding some players from the competition. In particular, for some game instances, it can be beneficial to exclude the highest-skill player. Intuitively, such exclusion may result in a more intense competition among players with more balanced skills, and, as a result, yield a higher expected total effort.

We now move on to discuss the game with incomplete information that models the standard all-pay contest. We restrict our discussion to prior distributions according to which skills of players are independent and identically distributed random variables. The game has a unique symmetric Bayes-Nash equilibrium, in which players play identical strategies. The expected total effort in this equilibrium is equal to the expected value of the second-highest skill of a player. Interestingly, the expected maximum individual effort is at least half of the expected total effort in any symmetric Bayes-Nash equilibrium, which was established by Chawla, Hartline, and Sivan.[7] This is exactly the same relation we previously noted to hold between the expected total effort and the expected maximum individual effort in any mixed-strategy Nash equilibrium of the game with complete information.

Rank-order allocation of prizes. We now consider a more general situation where a prize budget can be arbitrarily split among two or more position prizes, which are assigned to players in decreasing order of effort, subject to the constraint that any position prize is at least as large as any lower position prize (see Figure 3). For example, a prize budget may be split between two position prizes such that 2/3 of the prize budget is allocated to first place prize and the remaining part is allocated to second place prize; a common way of splitting a prize budget in TopCoder contests.

We consider the question of how should a prize budget be split among position prizes to maximize a given objective in equilibrium. Clearly, the answer depends on the choice of the objective, equilibrium concept, heterogeneity of skills, and production costs. Suppose the objective is to maximize the expected total effort in equilibrium of the game with incomplete information, where players have identical prior distributions of skills and unit marginal production costs. Under these assumptions, it is optimal to allocate the entire prize budget to first place prize, which was shown by Moldovanu and Sela.[28] Under the same assumptions, allocating the entire prize budget to the first-place prize is also optimal for the objective of maximizing the expected maximum individual effort, which was shown by Chawla, Hartline, and Sivan.[7] These results hold even more generally for any production cost function with decreasing marginal costs. In contrast, for production cost functions with increasing marginal costs, it may be optimal to split a prize budget among two or more position prizes. The optimality of allocating entire prize budget to first place prize holds also for the game with complete information, under the assumption that players have identical skills and decreasing marginal production costs, as shown by Glazer and Hassin[17] and Ghosh and McAfee.[16]

The assumption that in the game with incomplete information the skills of players have identical prior distributions is critical for the optimality of allocating entire prize budget to first place prize. Similarly, the assumption that in the game with complete information the skills of players are identical is critical for the optimality of allocating entire prize budget to first place prize. If the skills of players have non-identical prior distributions, then there exist game instances such that it is profitable to split the prize budget over two or more position prizes, which is shown by the following example.

Three players, two prizes example. Consider a game where a unit prize budget is split between two position prizes such that $\frac{1}{2} \leq \alpha \leq 1$ is allocated to the first place prize and the remaining part is allocated to the second place prize. Assume there are three players: a high-skill player with the skill parameter of value $v>1$ and two low-skill players whose skill parameters are of value 1. Assume that each player incurs unit marginal production cost. This game has a mixed-strategy Nash equilibrium such that the two low-skill players play symmetric strategies. This equilibrium is such that in the limit of asymptotically large skill of the high-skill player, the mixed strategy of the high-skill player converges to a uniform distribution on $[1 - \alpha,\alpha]$, and that of low-skill players converges to a uniform distribution on $[0,1 - \alpha]$. In this limit, the expected effort of the high-skill player is $\frac{1}{2}$, and that of each low-skill player is $(1 - \alpha)/2$. This adds up to the expected total effort of value $^{3}/_{2} - \alpha$. Therefore, we observe the more balanced the split of the prize budget between the two position prizes, the larger the expected total effort.

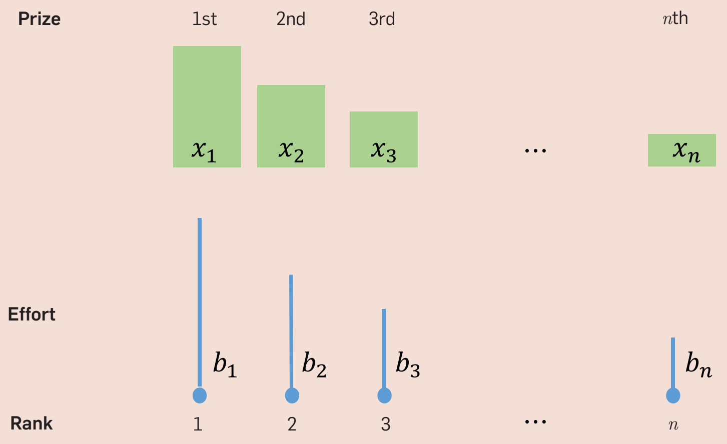
An interesting question to ask is how should a prize budget be allocated to maximize a given objective in equilibrium, without making a commitment to allocate the entire prize budget to players, no matter what effort investments they make. This question has been resolved for the game with incomplete information and the objective of maximizing the expected total effort by the celebrated work of Myerson.[28] In particular, if the skill parameters are independent and identically distributed according to a prior distribution that satisfies a certain regularity condition, it is optimal to award the entire prize budget to a highest-effort player subject to his or her effort being larger than or equal to a minimum required effort, and withhold the prize by the contest owner, otherwise. Chawla, Hartline, and Sivan[7] have recently established similar characterization of the optimum prize allocation for the objective of maximizing the expected maximum individual effort.

Smooth allocation of prizes. Now consider prize allocation mechanisms that have a positive bias to awarding players who invest high effort, but do not guarantee that the prize is allocated to a highest-effort player. Such prize allocation mechanisms can arise due to various factors. One factor is the stochasticity of production, where individual production outputs are random variables, positively correlated with invested efforts. Another factor is allocation of prizes based on a ranking of players derived from noisy observations of individual production outputs. Such prize allocation mechanisms are referred to be with *imperfect discrimination*. The stochasticity of production may result in prize allocation according to a smooth function of invested efforts, for all vectors of efforts except for some corner cases such as when all players invest zero efforts.

An example of a smooth allocation of prizes is *proportional allocation* that splits a prize budget among players in proportion to invested efforts, conditional on at least one player investing a strictly positive effort; otherwise, the prize is evenly split among players (Figure 4). A smooth prize allocation may be enforced by the design of a resource allocation mechanism. For example, proportional allocation has been used for allocation of computing resources[37] and network bandwidth.[22] Such resources typically consist of a large number of small units and, thus, for any practical purposes, can be regarded as infinitely divisible resources.

A more general class of smooth allocations is defined by allocating in proportion to an increasing positive-valued function of invested effort, referred to as a *general logit allocation*. A special case is allocation in proportion to a power function of invested effort, with a positive exponent parameter *r*. This is commonly referred to as *Tullock allocation*, which has been studied extensively in the literature on rent-seeking contests.[40] Proportional allocation is a special case of a Tullock allocation for the value of parameter *r* equal to 1. The larger the value of parameter *r*, the larger the share of

Figure 4. Proportional allocation.



$$x_i(\boldsymbol{b}) = \begin{cases} \dfrac{b_i}{\sum_{j=1}^{n} b_j}, & \text{if } \displaystyle\sum_{j=1}^{n} b_j > 0 \\ \dfrac{1}{n}, & \text{if } \displaystyle\sum_{j=1}^{n} b_j = 0 \end{cases}$$

the prize allocated to a highest-effort player. For more details about smooth allocations, see, for example, Corchon and Dahm[10] and Vojnović.[42]

One may ask how do equilibrium outcomes in the game that models the standard all-pay contest compare with those in the game with a smooth prize allocation, say, according to proportional allocation. A first notable difference is that unlike the game that models the standard all-pay contest, the game with proportional allocation has a pure-strategy Nash equilibrium, which is unique.

The total effort in any pure-strategy Nash equilibrium is guaranteed to be at least $v_2/2$. The total effort increases in the highest-skill parameter v_1 and it can be larger than v_2. This is in contrast to the game that models the standard all-pay contest, where the total effort in any mixed-strategy Nash equilibrium is at most v_2. One may ask whether there exists a smooth allocation of prizes that guarantees the total effort to be within a constant factor of v_1 in any pure-strategy Nash equilibrium. The answer is negative.[41] This gives us a useful insight that randomized prize allocations can achieve a larger total effort, but there are fundamental limits that cannot be surpassed.

The maximum individual effort can be an arbitrarily small fraction of the total effort; for example, this is so for the simple game instance with equally skilled players by taking the number of players to be sufficiently large. This is in contrast to the game that models the standard all-pay contest where we noted that in any mixed-strategy Nash equilibrium, the expected maximum individual effort is at least 1/2 of the expected total effort.

The social welfare in any pure-strategy Nash equilibrium of the game with proportional allocation is always at least 3/4 of the optimum value, a result by Johari and Tsitsiklis.[21] It has been shown the game with proportional allocation is a smooth game (for example, see Roughgarden[34]), which implies that the expected social welfare is at least 1/2 of the optimum value in any mixed-strategy Nash equilibrium.

Unlike the game that models the standard all-pay contest, the exclusion principle does not hold for the game that models the contest with proportional allocation.[14] For the game that models the contest with proportional allocation, the total effort in the pure-strategy Nash equilibrium cannot be increased by excluding some of the players from competition.

An interesting question to ask is how should a prize budget be allocated to maximize a given objective in equilibrium, without making a commitment to allocate the entire prize budget to players, no matter what effort investments they make.

Simultaneous contests. In the context of online services, a contest is often run simultaneously with other contests. For example, in competition-based crowdsourcing services, there are typically many open contests at any given time. Similarly, in online labor marketplaces, there are usually many open jobs at any given time. Multiple open contests provide players with alternative options to invest efforts, which can have a significant effect on the effort invested in any given contest. A player can invest effort only in a limited number of contests over a period of time, or he or she has a limited effort budget to invest over available contests. A worker may only be able to produce a high-quality work by focusing to a small number of projects at any given time, or he or she may only be able to devote a limited number of work hours per week. Game theory provides us with a framework to study the relation between the values of prizes offered by different contests and the effort investments across different contests in a strategic equilibrium.

We consider games that model *simultaneous standard all-pay contests* that offer prizes of arbitrary values. Such games have been studied for different types of production costs. We first consider the case where production costs are such it is feasible for each player to participate in at most one contest, in which he or she incurs a unit marginal production cost. In such a game, strategic decision making of a player consists of two components: choosing in which contest to invest effort, and deciding how much effort to invest in the chosen contest. This strategic decision making is informed by the available information, which consists of the values of prizes offered by different contests and the prior information about the skills of players. We consider the game with incomplete information, where the skill parameters of players are independent and identically distributed according to a prior distribution. This game has a symmetric Bayes-

Nash equilibrium, which admits an explicit characterization, established in DiPalantino and Vojnović.[11] In this equilibrium, there is a *segregation* of players in different skill levels, such that the players of the same skill level choose contests according to identical mixed strategies. A player of a higher skill level chooses a contest to participate from a smaller set of contests that offer highest prizes. A higher expected participation is attracted by contests that offer high prizes, according to a relation that exhibits diminishing returns with respect to the values of the prizes.

Another type of production costs is when each player is endowed with an effort budget that he or she can split arbitrarily over available contests. This game is closely related to so-called *Colonel Blotto game*: there are two colonels and two or more battlefields; each colonel is endowed with a number of troops that are simultaneously deployed over the battlefields; a battlefield is won by the colonel who places a larger number of troops on this battlefield, and the game is won by the colonel who wins more battlefields. A *continuous Colonel Blotto game* assumes that each colonel is endowed with an infinitely divisible amount of army force.

The game with players endowed with effort budgets has a rich set of equilibrium properties. There are game instances with a continuum of mixed-strategy Nash equilibria. For example, this is the case for the game with two players that have non-identical effort budgets and two or more standard all-pay contests that offer identical prizes. When players have identical effort budgets, the game has both pure and mixed-strategy Nash equilibria in which each player invests all his or her effort in one contest, provided that the number of players is sufficiently large. In the limit of many players, the equilibrium participation of players across different contests is proportional to the values of prizes.

Games that model simultaneous contests with players endowed with effort budgets have also been studied for other prize allocation mechanisms, including proportional allocation and equal-share allocation. The game that models simultaneous contests with proportional allocation and players endowed with effort budgets is not guaranteed to have a pure-strategy Nash equilibrium. A sufficient condition for the existence of a pure-strategy Nash equilibrium is that each contest has at least two players with strictly positive skill parameters. The social efficiency in a pure-strategy Nash equilibrium can be arbitrarily low in a worst case.

Sharing of the utility of production. There have been various studies of production systems where agents invest effort in one or more activities, which results in a utility of production that is shared among contributors according to a utility sharing mechanism. Some online services rely on user contributions and award credits to incentivize contributions. For example, some online services rely on user-generated content, such as questions and answers in online Q&A services, and award credits in terms of attention or reputation points, which are commensurate to user contributions. Sharing the utility of production has been also studied in the context of cognitive labor and allocation of scientific credit, for example, Kitcher[23] and Kleinberg and Oren.[24]

A central question here is about the social efficiency of production in strategic equilibrium outcomes. Several factors can contribute to social inefficiency of production, including the choice of the utility sharing mechanism, the nature of the utility of production functions, and the nature of production cost functions. Special attention has been paid to *local utility sharing mechanisms*, which specify the shares of the utility of production associated with a project exclusively based on the effort investments in this project, and not on the effort investments in other projects. It is of interest to understand social efficiency of *simple* local utility sharing mechanisms, for example, allocating a priori fixed shares of the utility of production in decreasing order of individual contributions or allocating in proportion to individual contributions.

The nature of the utility of production is a critical factor for the social efficiency of equilibrium outcomes. If the utility of production is allowed to be according to a non-monotonic function of effort investments, then there are game instances for which the utility of production in a pure-strategy Nash equilibrium is an arbitrarily small fraction of the optimum; for example, this can be for a single project game with proportional allocation. This is an instance of a general phenomenon known as the *tragedy of the commons*,[19] referring to an inefficient use of congestible resources that arises from non-cooperative behavior of selfish agents. The nature of the production cost functions is also a critical factor. If, in a single project game with proportional allocation, the utility of production is a monotone function, but players incur unit marginal production costs, then a similar inefficiency of production can arise.

Are there conditions for the games under consideration under which equilibrium is guaranteed to exist and all equilibria are approximately socially efficient? Here we may settle for the utility of production to be at least a constant factor of the optimum value. Such conditions have been identified by Vetta[41] for the class of games referred to as *monotone valid utility games*. A game is said to be a monotone valid utility game if the players' payoffs are according to utility functions whose sum is less than or equal to the value of a social utility function, and the following two conditions hold. The game is required to satisfy a *monotonicity condition*, which restricts to social utility functions whose value cannot increase by some player opting out from participation. The game is also required to satisfy a *marginal contribution condition*, which restricts each player's utility to be at least as large as his or her marginal contribution to the social utility. In the context of games that model simultaneous projects, whether or not the marginal contribution condition holds depends on the nature of the utility of production functions and the utility sharing mechanism. For example, the marginal contribution condition holds if the project utility functions are increasing functions with diminishing returns in the total effort invested in a project, and the utility sharing is according to proportional allocation. For monotone valid utility games, the utility of production in any pure-strategy Nash equilibrium is guaranteed to be at least 1/2 of the optimum value.

The approximate social efficiency of the utility of production in any pure-strategy Nash equilibrium has been established under the assumption that project utility functions have diminishing returns. The diminishing returns of the utility of production are representative of production systems in which individual contributions are substitutes. If, on the other hand, individual contributions are complements (that is, the utility of production has increasing returns), then the social efficiency in an equilibrium outcome can be arbitrarily low. In such cases, the utility of production in a pure-strategy Nash equilibrium cannot be guaranteed to be a constant-factor of the optimum value, but it is always at least $1/k$ of the optimum value, where k is the maximum number of players participating in a project.

Sequential contests and tournaments. So far we discussed games that model contests where players simultaneously invest effort. A variety of games have been studied that model contests with some elements of sequential play. A coverage of these games and related work is available.[42] Here we only mention some of these games: a single contest with sequential effort investments; a multi-round two-player contest where the winner is the player who first wins a given number of rounds more than the opponent, referred to as *tug-of-war*; a contest in which each player continuously invests effort until dropping out and the contest ends as soon as the number of players that are still in the competition is equal to the number of available prizes, referred to as *war-of-attrition*;[5] a multi-round contest that ends as soon as the utility of cumulative effort exceeds a threshold whose value is private information of the contest owner;[35] and, a contest where prizes are allocated over multiple rounds and each player competes until he or she wins a prize.[8]

Common contest architecture has the form of *a single-elimination tournament*, defined by a directed *tree* and a *seeding* of players. Each contest of the tournament has one winner and all players who lose in a contest are eliminated from further competition. The winner of the tournament is the player who wins all contests in which he or she participates. A typical single-elimination tournament consists of two-player contests and is defined by a binary tree and a seeding of players. Seeding procedures have been studied with respect to various criteria, such as the winning probability of the highest-skill player. These studies have been pursued under two different assumptions: contest outcomes are assumed to be independent random events according to given winning probabilities; and in each round of the tournament, the players who participate in this round make strategic effort investments accounting for their prospective payoffs in subsequent rounds of the tournament.

Skill-Rating Methods

An important component of some online services is a skill-rating system that uses as input observed contest outcomes. For example, a contest outcome may be a full ranking, that is, an ordered list of participants in the contest in decreasing order of individual performance, or a partial ranking such as a top-1 list that contains information about who participated in a contest and who was the winner in this contest. The skill ratings are used for various purposes, such as for creation of league tables, leaderboards, seeding of tournaments, and matchmaking in online labor platforms. Popular skill-rating systems include TrueSkill, used in online gaming,[20] TopCoder skill-rating system, and skill-rating systems used in various sport competitions.[13]

A common requirement for skill-rating systems is to allow for prediction of contest outcomes. For example, such predictions are used in online games for the purpose of matching equally skilled players, which results in interesting matches with uncertain outcomes. The design of skill-rating systems is often required to be based on simple and easy to understand principles, which are often made public information. The skill-rating systems often use only a few parameters to represent an individual's skill; for example, using a scalar parameter for a point estimate and one extra parameter for the uncertainty of the estimate.

Statistical models of ranking outcomes. The design of skill-rating systems is based on statistical models of ranking outcomes introduced by statisticians as early as in 1920s. A commonly used statistical model of ranking outcomes was introduced by Thurstone.[39] Under this model, each comparison of a given set of individuals results in a ranking of these individuals generated as follows. The individuals are associated with latent performance random variables that are assumed to be independent across different individuals and comparisons. The ranking outcome of a comparison is assumed to be in decreasing order of individual performance. Each individual performance is equal to a deterministic skill parameter plus a zero mean noise random variable. The value of the skill parameter is unknown and has to be inferred from the observed ranking outcomes. The noise random variables are assumed to be independent and identically distributed over different individuals and different comparisons.

Specifically, for a comparison of a set S of individuals, each individual $i \in S$ is associated with performance $b_i = v_i + \varepsilon_i$, where v_i is a real-valued skill parameter and ϵ_i is a zero mean noise random variable. A ranking outcome is derived from admitting that i is ranked higher than j whenever their respective performances satisfy $b_i > b_j$.

A common assumption is that noise random variables are according to a Gaussian distribution, with zero mean and known variance β^2. This assumption was made in the original work by Thurstone for pair comparisons, and has been admitted by many popular skill-rating systems, including TrueSkill, TopCoder skill-rating system, and skill-rating systems used in various sport competitions. The probability that individual i is ranked higher than individual j, in a comparison that involves these two individuals, is given by

$$p_{i,j} = \mathbf{P}\left[b_i > b_j\right] = \Phi\left(\frac{1}{\sqrt{2}\beta}(v_i - v_j)\right)$$

where $\Phi(\cdot)$ is the cumulative standard normal distribution.

An important component of some online services is a skill ratings system that uses as input observed contest outcomes.

Another well-known model is the Bradley-Terry model, first introduced by Zermelo in 1920s[45] and later popularized in 1950s by the work of Bradley and Terry[4] and others. Under the Bradley-Terry model, the probability that individual i is ranked higher than individual j, in a comparison that involves these two individuals, is given by

$$p_{i,j} = \frac{\theta_i}{\theta_i + \theta_j}$$

where θ_i and θ_j are positive-valued skill parameters. According to the Bradley-Terry model, the winning probability of an individual in a pair comparison with another individual is proportional to his or her skill parameter. The natural generalization to comparison sets of two or more individuals, where the winning probabilities are proportional to the skill parameters, is known as the Luce's choice model. Another generalization is a model of full ranking outcomes for comparison sets of two or more individuals, defined by sampling individuals from a given comparison set without replacement with probabilities proportional to their skill parameters; this is known as the Plackett-Luce model. The Luce's choice model is a special instance of a Thurstone model with noise random variables according to a double-exponential distribution with zero mean and variance β^2. In this case, we have

$$p_{i,j} = \mathbf{P}[b_i > b_j] = \frac{1}{1 + e^{-\frac{\pi}{\sqrt{6}\beta}(v_i - v_j)}}$$

that corresponds to the Bradley-Terry model by using the change of parameters $v_i = \sqrt{6}\beta/\pi \log(\theta_i)$.

The statistical models of ranking outcomes discussed so far have been extended to accommodate various requirements of modern applications. For example, they have been extended to allow for skill rating based on observed outcomes of team competitions, which arises in online gaming applications. This extension is based on a model that assumes a team performance to be according to a given function of individual performances. For instance, a team performance may be assumed to be a linear function of individual performances, such as in the TrueSkill rating system. An area in which advances have been made is on statistical inference methods, which we briefly review as follows.

Statistical inference methods. Having admitted a statistical model of ranking outcomes, it remains to choose a statistical inference method for estimation of skill parameters based on observed ranking outcomes. Two approaches are in common use: a frequentist approach and a Bayesian approach. The frequentist approach considers skill parameters as unknown parameters and estimates them by minimizing a given loss function, for example, the negative log-likelihood in the case of the maximum likelihood estimation. The Bayesian approach considers skill parameters as random variables with a given prior distribution, and amounts to computing the posterior distribution of these random variables conditional on the observed ranking outcomes.

Frequentist inference. Statistical models of pair comparisons, such as the Thurstone model with either Gaussian or double-exponential distribution of noise, have a unique maximum likelihood estimator (up to an additive constant) provided that the adjacency matrix, specifying how many times different pairs of individuals are compared in the input data, is irreducible. An *adjacency matrix* is said to be irreducible if the corresponding graph, we refer to as a comparison graph, is connected. It was recently shown that the accuracy of the maximum likelihood parameter estimator critically depends on how well the comparison graph is connected, for example, Hajek, Ox and Xu[18] and Vojnovic and Yun.[43] Specifically, a key parameter is the *algebraic connectivity* of the comparison graph, defined as the second smallest eigenvalue of the Laplacian matrix of the comparison graph. Another line of recent research is on various iterative methods for skill parameter estimation, including gradient-descent based methods for minimizing the negative log-likelihood function, as well as alternative methods based on spectral properties of matrices and random walks, for example, Neghaban, Oh, and Shah.[30]

Bayesian inference. For statistical models of ranking outcomes according to a Thurstone model, the posterior distribution of an individual's skill is a marginal distribution of the posterior joint distribution of a multivariate variable that consists of individual skills and individual performances. This posterior joint distribution consists of several factors that are conveniently represented by a graphical model, a way to represent the information about which factors depend on which variables. The marginal posterior distributions of skills can be computed using standard message-passing methods for inference in graphical models, such as the sum-product algorithm. It is common to approximate a marginal distribution of a skill variable with a distribution from an assumed family of distributions; for example, assuming the family of Gaussian distributions, as done in the TrueSkill rating system. The approximate Bayesian inference amounts to approximating marginal posterior distributions of skills by distributions from the given family of distributions, assuming that marginal prior distributions belong to this family of distributions.

Future Directions

Strategic game models of contests provide plenty of interesting hypotheses about what strategic user behavior may arise in different contest situations. Future work must be devoted to narrowing the gap between theoretical results and empirical validations. The availability of online services whose design is based on contests and the collected data provides us with an opportunity to test the existing theories and guide the development of new contributions to contest theory. Another research direction is to study statistical inference methods for various contest designs, such as in the recent study of A/B testing for auctions.[6]

While the skill-rating methods have been studied extensively over many years, some interesting questions still remain open. Most skill-rating methods represent an individual's skill by a scalar parameter. In many situations, however, it is of interest to consider an individual's skill over multiple dimensions; for example, an online worker may have different types of skills such as analytical problem solving, strategic business planning, and software programming skills. Another interesting direction is to study statistical inference methods for statistical models of ranking outcomes that allow for a larger set of unknown parameters. For example, in an online labor platform, a ranking of job applicants would depend not only on the idiosyncratic skills of the applicants, but also on the specific job requirements, both of which may have uncertainties. Another direction is to develop solid theoretical foundations for individual skill rating based on observed team performance outputs. Current statistical inference methods used in practice assume simple models of team performance, such as that a team performance is the sum of individual performances, which may not always be valid in practice.

References

1. Archak, N. Money, glory and cheap talk: analysing strategic behavior of contestants in simultaneous crowdsourcing contests on TopCoder.com. In *Proceedings of WWW '10* (Raleigh, N.C., 2010), 21–30.
2. Baye, M.R., Kovenock, D. and de Vries, C.G. The all-pay auction with complete information. *Econ. Theory 8*, 2 (1996), 291–305.
3. Bishop, D.T. et al. The war of attrition with random rewards. *J. Theoretical Bio 70*, 1 (1978), 85–124.
4. Bradley, R. A. and Terry, M.E. Rank analysis of incomplete block designs: I. Method of paired comparisons. *Biometrika 20*, 3–4 (1952), 334–345.
5. Bulow, J. et al. The generalized war of attrition. *Amer. Econ. Rev. 39*, 3–4 (1997), 324–345.
6. Chawla, S., Hartline, J.D. and Nekipelov, D. A/B testing of auctions. In *Proceedings of ACM EC '16* (Maastricht, Neterlands, 2016), 856–868.
7. Chawla, S., Hartline, J.D. and Sivan, B. Optimal crowdsourcing contests. In *Proceedings of SODA '12*, (Kyoto, Japan, 2012), 856–868.
8. Clark, D.J. and Riis, C. Competition over more than one prize. *Amer. Econ. Rev. 88*, 1 (1998), 276–289.
9. Corchon, L.C. The theory of contests: A survey. Rev. *Econ. Design 11* (2007), 69–100.
10. Corchon, L. and Dahm, M. Foundations for contest success functions. *Econ.Theory 88*, 1 (2010), 81–98.
11. DiPalantino, D. and Vojnovic, M. Crowdsourcing and all-pay auctions. In *Proceedings of ACM EC '09* (Stanford, CA, 2009), 119–128.
12. Doan, A., Ramakrishnan, R. and Halevy, A.Y. Crowdsourcing systems on the World-Wide Web. *Commun. ACM 54*, 4 (Apr. 2011), 85–96.
13. Elo, A.E. *The rating of chessplayers.* Ishi Press International, 1978.
14. Franke, J., Kanzow, C., Leininger, W. and Schwartz, A. Lottery versus all-pay auction contests: A revenue dominance theorem. *Games and Economic Behavior, 13* (2014), 116–126.
15. Galton, F. The most suitable proportion between the value of first and second prizes. *Biometrika 1*, 4 (1902), 385–399.
16. Ghosh, A. and McAfee, R.P. Crowdsourcing with endogenous entry. In *Proceedings of WWW '12* (Lyon, France, 2012), 999–1008.
17. Glazer, A. and Hassin, R. Optimal contests. *Economic Inquiry 26*, 1 (1988), 133–143.
18. Hajek, B., Oh, S. and Xu, J. Minimax-optimal inference from partial rankings. In *Proceedings of NIPS '14*, (Montreal, Quebec, 2014), 1475–1483.
19. Hardin, G. The tragedy of the commons. *Science* 162, 3859 (1968), 1243–1248.
20. Herbrich, R., Minka, T. and Graepel, T. TrueSkill: A Bayesian skill rating system. In *Proceedings of NIPS '06*, (Vancouver, B.C., 2006), 569–576.
21. Johari, R. and Tsitsiklis, J.N., Efficiency loss in a network resource allocation game. *Math. Operations Res 29*, 3 (2004), 402–435.
22. Kelly, F. Charging and rate control for elastic traffic. *European Trans. Telecommun. 8*, 1 (1997), 33–37.
23. Kitcher, P. The division of cognitive labor. *J. Philosophy 87*, 1 (1990), 5–22.
24. Kleinberg, J. and Oren, S. Mechanisms for (mis) allocating scientific credit. In *Proceedings of STOC '11* (San Jose, CA, 2011), 529–538.
25. Konrad, K.A. Strategy in Contest—An Introduction. *WZB-Markets and Politics Working Paper N. SP II 2007-01*; 2007 (http://ssrn.com/abstract=960458).
26. Krueger, A.O. The political economy of the rent-seeking society. *Amer. Econ. Rev. 64*, (1974), 291–303.
27. Lazear, E.P. and Rosen, S. Rank-order tournaments as optimum labor contracts. *J. Pol. Econ. 89*, 5 (1981), 841–864.
28. Moldovanu, B. and Sela, A. The optimal allocation of prizes in contests. *American Econ. Rev. 91*, 3 (2001), 542–558.
29. Myerson, R.B. Optimal auction design. *Mathematics of Operations Research 6*, 1 (1981), 58–73.
30. Neghaban, S., Oh, S., and Shah, D. Iterative ranking from pair-wise comparisons. In *Proceedings of NIPS '12*, (Lake Tahoe, NV, 2012), 2483–2491.
31. Nisan, N., Roughgarden, T., Tardos, E. & Vazirani, V. V., 2007. *Algorithmic Game Theory.* Cambridge University Press.
32. Nitzan, S., 1994. Modelling rent-seeking contests. *Eur. J. Polit. Econ. 10* (1994),, pp. 41-60.
33. Roughgarden, T. Algorithmic game theory. *Commun. ACM 53*, 7 (2010), 78–86.
34. Roughgarden, T. Intrinsic robustness of the prize of anarchy. *Commun. ACM 55*, 7 (2012), 116–123.
35. Shaili, J., Yiling, C., Parkes, D.C. Designing incentives for online question and answer forums. In *Proceedings of ACM EC '09* (Stanford, CA, 2009), 129–138.
36. Shoham, Y. Computer science and game theory. *Commun. ACM 51*, 8 (2008), 75–79.
37. Stoica, I. et al. A proportional share resource allocation algorithm for real-time, time-shared systems. In *Proceedings of the 17th Real-Time Systems Symposium*. (Washington, D.C., 1996), 288–299.
38. Tang, J.C. et al. Reflecting on the DARPA Red Balloon Challenge. *Commun. 54*, 4 (2011), 78–85.
39. Thurstone, L.L. A law of comparative judgment. *Psychological Review 34*, 2 (1927), 273–286.
40. Tullock, G., Efficient rent seeking. In *Theory of the Rent-Seeking Society*. A&M University Press (1980), 131–146.
41. Vetta, A., *Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions.* In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science* (2002), 416–425.
42. Vojnović, M. *Contest Theory: Incentive Mechanisms and Ranking Methods.* Cambridge University Press, 2016.
43. Vojnović, M. and Yun, S.-Y., Parameter estimation for generalized Thurstone choice models. In *Proceedings of ICML '16* (New York City, NY, 2016).
44. Yang, J., Adamic, L., and Ackerman, M. Crowdsourcing and knowledge sharing: Strategic user behavior on Taskcn. In *Proceedings of the ACM EC '09* (Chicago, Il, 2008).
45. Zermelo, E. Die Berechnung der Turnier-Ergebnisse al sein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift 29* (1929), 436–460.

Milan Vojnović (m.vojnovic@lse.ac.uk) is a professor of data science in the Department of Statistics, London School of Economics, U.K., where he serves as program director of MSc in Data Science.

Watch the author discuss his work in this exclusive *Communications* video. http://cacm.acm.org/videos/contest-theory

research highlights

DOI:10.1145/3060595

To view the accompanying paper, visit doi.acm.org/10.1145/3060597

Technical Perspective
Functional Compilers

By Guy Blelloch

The following paper points out that stream fusion by itself is not well suited for generating bulk instructions such as vector or SIMD instructions.

PROGRAMMING IN A functional programming style can often lead to surprisingly elegant solutions to complicated problems. This arises in part from abstracting away from locations and state and thinking instead in terms of values and functions, in a mathematical style. Also, importantly, the lack of side effects means that the components are easily composable. This is particularly important for parallel programs since it means the lack of side effects leads to code that can run in parallel but has a deterministic sequential semantics. Since the functional programming style focuses on values rather than state, it abstracts away from the notion of memory and location. This can be viewed as a failure, or as an opportunity.

On the one side it fails to let the user control how memory is laid out or how operations are ordered during the computation. This disallows many optimizations by the user that are crucial for performance on modern hardware—for example, laying out structures adjacently so they share a cache line, or avoiding levels of indirection, often referred to as boxing.

On the other side it is an opportunity for smart compilers or runtime systems to do these optimizations for the user. The compiler has the advantage that it can be customized for different machines, and can potentially have a more accurate model of the costs of a machine. Also compilers are more capable of searching large parameter spaces—it is surely rare that any humans still do register allocation by hand. On this side, compilers for typed functional languages have taken large steps at generating code that can sometimes match or even beat optimized low-level human generated codes. Such compilers include the MLton compiler for Standard ML and the Glasgow Haskell Compiler (GHC) for Haskell. Both are very proficient at unboxing and hence avoiding levels of indirection. GHC also performs stream fusion, which can avoid generating intermediate results that are expensive to write and read back. The following paper by Mainland, Leshchinskiy, and Peyton Jones points out, however, that stream fusion by itself is not well suited for generating bulk instructions such as vector or SIMD instructions.

As an example, the authors consider a simple vector dot product. A dot product is expressed naturally, and compositionally, as an element-wise product of the two vectors, followed by a sum of the elements of the resulting vector—or in functional parlance, a zip-with multiply followed by a reduce plus. This is elegant and high-level because it does not directly specify the ordering of how the element-wise multiplies or sums in the reductions are applied.

Naïvely, however, such a dot product creates an intermediate vector containing all the element-wise products. This is inefficient since writing out the intermediate vector and reading it back will end up being a significant portion of the cost. Instead it can be much more efficient to multiply a pair and immediately add it into the running sum, as one would likely write in a loop using an imperative language such as C or Java. The translation from the zip-reduce solution to such a loop form can be done automatically by the Haskell compiler using stream fusion. However, the resulting code is inherently sequential, as would be the C or Java code, and inhibits the use of bulk operations, or vector instructions. Instead, the target code needs to be able to chunk (or block) the vectors into pieces to which the bulk operations or vector instructions can be applied.

The authors propose a solution for such chunking. The approach recognizes that no one representation is useful for all situations, so instead maintains multiple representations of a stream in what they refer to as a bundle. Maintaining multiple representations might seem inherently inefficient due to redundancy, but given the stream framework, only one representation need be generated for a producer at the behest of the consumer, and the unevaluated remaining ones can be tossed. Making this all work imposes several other challenges that are discussed. Ultimately, the paper provides a variety of results that show the approach can lead to Haskell code outperforming C on certain benchmarks even when it uses the vector library.

The holy grail of compilers for functional languages, that is, always outperforming hand-tuned code, has certainly not yet been achieved in general, but compilers for typed functional languages continue to make big steps. C

Guy Blelloch is a professor of computer science at Carnegie Mellon University, Pittsburgh, PA.

DOI:10.1145/3060597

Exploiting Vector Instructions with Generalized Stream Fusion

By Geoffrey Mainland*, Roman Leshchinskiy, and Simon Peyton Jones

Abstract

Ideally, a program written as a composition of concise, self-contained components should perform as well as the equivalent hand-written version where the functionality of what was many components has been manually combined into a monolithic implementation. That is, programmers should not have to sacrifice code clarity or good software engineering practices to obtain performance—we want compositionality without a performance penalty. This work shows how to attain this goal for high-level Haskell in the domain of sequence-processing functions, which includes applications such as array processing.

Prior work on stream fusion[3] shows how to automatically transform some high-level sequence-processing functions into efficient implementations. It has been used to great effect in Haskell libraries for manipulating byte arrays, Unicode text, and unboxed vectors. However some operations, like vector append, do not perform well within the stream fusion framework. Others, like SIMD computation using the SSE and AVX instructions available on modern x86 chips, do not seem to fit in the stream fusion framework at all. We describe generalized stream fusion, which solves these issues through a careful choice of stream representation. Benchmarks show that high-level Haskell code written using our compiler and libraries can produce code that is faster than both compiler- and hand-vectorized C.

1. INTRODUCTION

It seems unreasonable to ask a compiler to be able to turn numeric algorithms expressed as high-level Haskell code into tight machine code. The compiler must cope with boxed numeric types, handle lazy evaluation, and eliminate intermediate data structures. However the Glasgow Haskell Compiler has become "sufficiently smart" that, in many domains, Haskell libraries for expressing numerical computations no longer have to sacrifice speed at the altar of abstraction.

The key development that made this sacrifice unnecessary is *stream fusion*.[3] Algorithms over sequences—whether they are lists or vectors (arrays)—are expressed naturally in a functional language using operations such as folds, maps, and zips. Although highly modular, these operations produce unnecessary intermediate structures that lead to inefficient code. Eliminating these intermediate structures is termed deforestation, or fusion. Equational laws, such as $\mathsf{map}\ \mathsf{f} \circ \mathsf{map}\ \mathsf{g} \equiv \mathsf{map}\ (\mathsf{f} \circ \mathsf{g})$, allow some of these intermediate structures to be eliminated; finding more general rules has been the subject of a great deal of research.

Stream fusion, based on the observation that recursive structures can be transformed into non-recursive co-structures for which fusion is relatively straightforward, was the first truly general solution. Instead of working directly with lists or vectors, stream fusion works by re-expressing these functions as operations over streams, each represented as a state and a step function that transforms the state while potentially yielding a single value. Alas, different operations need different stream representations, and no single representation works well for all operations (Section 2.2). Furthermore, for many operations it is not obvious what the choice of representation should be.

We solve this problem with a new *generalized stream fusion* framework where the primary abstraction used to express operations on vectors is a *bundle* of streams. The streams are chosen so that for any given high-level vector operation there is a stream in the bundle whose representation leads to an efficient implementation. The bundle abstraction has no run-time cost because standard optimizations performed by the Glasgow Haskell Compiler (GHC) eliminate intermediate bundle structures. We describe the generalized stream framework as well as a stream representation that leads to efficient vectorized code. Our benchmarks compare to the very best C and C++ compilers and libraries that we could find. Remarkably, our benchmarks show that choosing the proper stream representations can result in machine code that beats compiler-vectorized C and is competitive with hand-tuned assembly.

2. BACKGROUND

We begin by providing the background necessary for understanding stream fusion. There is no new material here—it is all derived from Coutts et al.[3] However, we describe fusion for functions of *vectors* of unboxed values, as implemented in the `vector`[10] library, rather than fusion for functions over *lists*. Some of the implementation details are elided, but the essential aspects of stream fusion as we describe them are faithful to the implementation.

The big idea behind stream fusion is to rewrite recursive functions, which are difficult for a compiler to automatically optimize, as non-recursive functions. The abstraction that accomplishes this is the Stream data type:

```
data Stream a where
  Stream :: (s → Step s a) → s → Int → Stream a
data Step s a = Yield a s
              | Skip s
```

The original version of this paper was published in *Proceedings of the 18th SIGPLAN International Conference on Functional Programming* (Boston, MA, 2013), 37–48.

* This work was performed while the author was at Microsoft Research Ltd.

```
                  | Done
```

A stream is a triple of values: an internal (existentially quantified) state, represented by the type variable s in the above definition, a size, and a step function that, when given a state, produces a Step. A Step may be Done, indicating that there are no more values in the Stream, it may Yield a value and a new state, or it may produce a new state but Skip producing a value. The presence of Skip allows us to easily express functions like filter within the stream fusion framework.

To see concretely how this helps us avoid recursive functions, let us write map for vectors using streams

```
map :: (a → b) → Vector a → Vector b
map f = unstream ∘ map_s f ∘ stream
```

The functions stream and unstream convert a Vector to and from a stream. A Vector is converted to a stream whose state is an integer index and whose step function yields the value at the current index, which is incremented at each step. To convert a stream back into a Vector, unstream allocates memory for a new vector and writes each element to the vector as it is yielded by the stream—unstream embodies a recursive loop. Though imperative, the allocation and writing of the vector are safely embedded in pure Haskell using the ST monad.[9]

The real work is done by map$_s$, which is happily non-recursive:

```
map_s :: (a → b) → Stream a → Stream b
map_s f (Stream step s) = Stream step′s
  where
    step′ s = case step s of
      Yield x s′ → Yield (f x) s′
      Skip s′    → Skip s′
      Done       → Done
```

With this definition, the equational rule mentioned in the Introduction, map f ∘ map g ≡ map (f ∘ g), falls out automatically. To see this, let us first inline our new definition of map in the expression map f ∘ map g:

```
map f ∘ map g ≡
  unstream ∘ map_s f ∘ stream ∘ unstream ∘ map_s
    g ∘ stream
```

Given this form, we can immediately spot where an intermediate structure is formed—by the composition stream ∘ unstream. This composition is, in effect, the identity function, so we should be able to eliminate it entirely. GHC's rewrite rules enable programmers to express algebraic identities such as stream ∘ unstream = id in a form that GHC can understand and automatically apply. Stream fusion relies *critically* on this ability, and the `vector` library includes exactly this rule. With the rule in place, GHC transforms our original composition of maps into

```
map f ∘ map g ≡
  unstream ∘ map_s f ∘ map_s g ∘ stream
```

Conceptually, stream fusion pushes all recursive loops into the final consumer. The two composed invocations of map become a composition of two *non-recursive* calls to map$_s$. The inliner is now perfectly capable of combining map$_s$ f ∘ map$_s$ g into a single Stream function. Stream fusion gives us the equational rule map f ∘ map g ≡ map (f ∘ g) *for free*.

2.1. Fusing the vector dot product

The motivating example we will use for the rest of the paper is the vector dot product. A high-level implementation of this function in Haskell might be written as follows:

```
dotp :: Vector Double → Vector Double → Double
dotp v w = sum (zipWith (∗) v w)
```

It seems that this implementation will suffer from severe inefficiency—the call to zipWith produces an unnecessary intermediate vector that is immediately consumed by the function sum. In expressing dotp as a composition of collective operations, we have perhaps gained a bit of algorithmic clarity, but in turn we have incurred a performance hit.

We have already seen how stream fusion eliminates intermediate structures in the case of a composition of two calls to map. Previous fusion frameworks could handle that example but were stymied by the presence of a zipWith. However stream fusion has no problem fusing zipWith, which we can see by applying the stream transformations we saw earlier to dotp.

The first step is to re-express each Vector operation as the composition of a Stream operation and appropriate conversions between Vectors and Streams at the boundaries. The functions zipWith and sum are expressed in this form as follows:

```
zipWith :: (a → b → c) → Vector a → Vector b →
  Vector c
zipWith f v w = unstream (zipWith_s f (stream v)
  (stream w))
sum :: Num a ⇒ Vector a → a
sum v = foldl′_s 0 (+) (stream v)
```

It is now relatively straightforward to transform dotp to eliminate the intermediate structure:

```
dotp :: Vector Double → Vector Double → Double
dotp v w ≡ sum (zipWith (∗) v w)
         ≡ foldl′_s 0 (+) (stream (unstream
               (zipWiths (∗) (stream v) (stream w))))
         ≡ foldl′_s 0 (+)
               (zipWiths (∗) (stream v) (stream w))
```

This transformation again consists of inlining a few definitions, something that GHC can easily perform, and rewriting the composition stream ∘ unstream to the identity function. After this transformation, the production (by zipWith) and following consumption (by sum) of an intermediate Vector becomes the composition of non-recursive functions on streams.

We can see how iteration is once again pushed into the final consumer by looking at the implementations of foldl'$_s$ and zipWith$_s$. The final consumer in dotp is foldl′$_s$, which is implemented by an explicit loop that consumes stream values and combines the yielded values with the accumulator z using the function f (the call to seq guarantees that the accumulator is strictly evaluated):

```
foldl′_s :: (a → b → a) → a → Stream b → a
foldl′_s f z (Stream step s) = loop z s
  where
loop z s = z ‘seq’
  case step s of
    Yield x s′ → loop (f z x) s′
    Skip s′    → loop z s′
    Done       → z
```

However, in zipWith$_s$ there is no loop—the two input streams are consumed until either both streams yield a value, in which case a value is yielded on the output stream, or until one of the input streams is done producing values. The internal state of the stream associated with zipWith$_s$ contains the state of the two input streams and a one-item buffer for the value produced by the first input stream:

```
zipWith_s :: (a → b → c) → Stream a → Stream
  b → Stream c

zipWith_s f (Stream stepa sa na) (Stream stepb sb nb) =
    Stream step (sa, sb, Nothing) (min na nb)
  where
    step (sa, sb, Nothing) =
      case stepa sa of
        Yield x sa′ → Skip (sa′, sb, Just x)
        Skip sa′    → Skip (sa′, sb, Nothing)
        Done        → Done
    step (sa, sb, Just x) =
      case stepb sb of
        Yield y sb′ → Yield (f x y) (sa, sb′, Nothing)
        Skip sb′    → Skip (sa, sb′, Just x)
        Done        → Done
```

Given these definitions, GHC's call-pattern specialization in concert with the standard inliner suffice to transform dotp into a single loop that does not produce an intermediate structure. If there is any doubt that this results in efficient machine code, we give the actual assembly language inner loop output by GHC using the LLVM back end. Stream fusion preserves the ability to write compositionally without sacrificing performance:

```
.LBB2_3:
    movsd (%rcx), %xmm0
    mulsd (%rdx), %xmm0
    addsd %xmm0, %xmm1
    addq  $8, %rcx
    addq  $8, %rdx
    decq  %rax
    jne .LBB2_3
```

2.2. Stream fusion inefficiencies

Though stream fusion does well for the examples we have shown, it still does not produce efficient implementations for many other operations. In particular, the inadequacy of the single-value-at-a-time nature of streams becomes particularly problematic when attempting to opportunistically utilize the SIMD instructions available on many current architectures, for example, SSE on x86 and NEON on ARM. These instructions operate in parallel on data values that contains two (or four or eight, depending on the hardware architecture) floating point numbers at a time. To avoid notational confusion, we call these *multi-values*, or sometimes just *multis*.

To enable sum to use SIMD instructions, we would like a stream representation that yields multi-values (rather than scalars), with perhaps a bit of scalar “dribble” at the end of the stream when the number of scalar values is not divisible by the size of a multi.

Although a stream of scalar values is useless for SIMD computation, a stream of multi-values is not quite right either, because of the “dribble” problem. Perhaps, we could get away with a stream that yielded *either* a scalar or a multi at each step, but this would force all scalar-only operations to handle an extra case, complicating the implementations of *all* operations and making them less efficient. There is a better way!

3. GENERALIZED STREAM FUSION

We have seen that different stream operations work best with different stream representations. In this section, we describe how to incorporate multiple stream representations into the stream fusion framework, elaborate on the details of a representation that enables SIMD computation with vectors, and show how to use our framework to transparently take advantage of SIMD instructions in Data Parallel Haskell programs.

The idea underlying generalized stream fusion is straightforward but its effects are wide-ranging: instead of transforming a function over vectors into a function over streams, transform it into a function over a *bundle* of streams. A bundle is a collection of streams, each semantically identical but with a different cost model, allowing each stream operation to choose the most advantageous stream representation in the bundle. We give a simplified version of the Bundle data type here:

```
data Bundle a = Bundle
  {sSize    :: Size
  , sElems  :: Stream a
  , sChunks :: Stream (Chunk a)
  , sMultis  :: Multis a
  }
```

The sElems field of the Bundle data type contains the familiar stream of scalar values that we saw in Section 2. The stream of Chunks contained in the sChunks field of the record enables the efficient use of bulk memory operations, like vector append, which we do not describe here. We next describe the representation contained in the sMultis field of the record, which enables the efficient use of SSE instructions.

3.1. A stream representation fit for SIMD computation

Modifying the stream fusion framework to accommodate SIMD operations opens up the possibility of dramatically increased performance for a wide range of numerical algorithms but requires a more thoughtful choice of representation. We focus on SIMD computations using 128-bit wide vectors and SSE instructions on x86/x64 since that is what our current implementation supports, although the approach generalizes.

Our implementation represents SIMD values using the type family Multi. We have chosen the name to avoid confusion with the Vector type, which represents arrays of arbitrary extent. In contrast, a value of type Multi a is a short vector containing a *fixed* number of elements—known as its *multiplicity*—of type a. On a given platform, Multi a has a multiplicity that is appropriate for the platform's SIMD instructions. For example, on x86, a Multi Double, will have multiplicity 2 since SSE instructions operate on 128-bit wide vectors, whereas a Multi Float will have multiplicity 4. Multi is implemented as an associated type[1] in the MultiType type class; their definitions are shown in Figure 1. MultiType includes various operations over Multi values, such as replicating a scalar across a Multi and folding a function over the scalar elements of a Multi. These operations are defined in terms of new primitives we added to GHC that compile directly to SSE instructions.

Given a value of type Vector Double, how can we operate on it efficiently using SSE instructions within the generalized stream fusion framework? An obvious first attempt is to include a stream of Multi Doubles in the stream bundle. However, this representation is insufficient for a vector with an odd number of elements since we will have one Double not belonging to a Multi at the end—the "dribble" mentioned earlier. Let us instead try this instead: a stream that can contain *either* a scalar or a Multi. We call this stream a MultisP because the *producer* chooses what will be yielded at each step:

```
data Either a b = Left a | Right b
type MultisP  a = Stream (Either a (Multi a))
```

Now we can implement summation using SIMD operations. Our strategy is to use two accumulating parameters, one for the sum of the Multi values yielded by the stream and one for the sum of the scalar values. Note that (+) is overloaded: we use *SIMD* (+) to add summ and y, and *scalar* (+) to add sum1 and x:

```
msumP_s :: (Num a, Num (Multi a)) ⇒ MultisP a → a
msumP_s (Stream step s _) = loop 0.0 0.0 s
  where
    loop summ sum1 s =
      case step s of
        Yield (Left x)  s′ → loop summ (sum1 + x) s′
        Yield (Right y) s′ → loop (summ + y) sum1  s′
        Skip            s′ → loop summ       sum1  s′
        Done               → multifold (+) sum1 summ
```

When the stream is done yielding values, we call the multifold member of the MultiType type class to fold the addition operator over the components of the Multi.

Figure 1. The MultiType type class and its associated type, Multi.

```
class MultiType a where
  data Multi a  -- Associated type
     -- The number of elements of type a in a Multi a.
  multiplicity :: Multi a  → Int
     -- A Multi a containing the values 0, 1, ...,
     -- multiplicity - 1.
  multienum :: Multi a
     -- Replicate a scalar across a Multi a.
  multireplicate :: a→ Multi a
     -- Map a function over the elements of a Multi a.
  multimap :: (a  → a) → Multi a  → Multi a
     -- Fold a function over the elements of a Multi a.
  multifold :: ( b → a → b) → b → Multi a  → b
     -- Zip two Multi a's with a function.
  multizipWith  ::  (a → a → a)
                 → Multi a  → Multi a  → Multi a
```

This implementation strategy works nicely for folds. However, if we try to implement the SIMD equivalent of zipWith$_s$, we hit a roadblock. A SIMD version of zipWith$_s$ requires that at each step either *both* of its input streams yield a Multi or they *both* yield a scalar—if one stream were to yield a scalar while the other yielded a Multi, we would have to somehow buffer the components of the Multi. And if one stream yielded *only* scalars while the other yielded only Multis, we would be hard-pressed to cope.

Instead of a stream representation where the producer chooses what is yielded, let us instead choose a representation where the stream *consumer* is in control:

```
data MultisC a where
  MultisC  ::  (s → Step s (Multi a))
             → (s → Step s a)
             → s
             → MultisC a
```

The idea is for a MultisC a to be able to yield either a value of type Multi a or a value of type a—the stream consumer chooses, which by calling one of the two step functions. Note that the existential state is quantified over both step functions, meaning that the same state can be used to yield either a single scalar or a Multi. If there is not a full Multi available, the first step function will return Done. The remaining scalars will then be yielded by the second step function. This representation allows us to implement a SIMD version of zipWith$_s$.

Regrettably, a MultisC *still* is not quite what we need. Consider appending two vectors of Doubles, each of which contains 41 elements. We cannot assume that the two vectors being appended are laid out consecutively in memory, so even though the stream that results from appending them together will contain 82 scalars, this stream is forced to yield a scalar in the middle of the stream. One might imagine an implementation that buffers and shifts partial Multi values, but this leads to very inefficient code. The alternative is for append$_s$ to produce a stream in which either a scalar or a Multi is yielded at each step—but that was the original representation we selected and then discarded because it was not suitable for zips!

The final compromise is to allow either—but not both—of these two representations. We cannot allow both—hence there is only one new bundle member rather than two—because while we can easily convert a MultisC a into a MultisP a, the other direction is not efficiently implementable. The final definition of the Multis type alias is therefore:

```
type Multis a = Either (MultisC a) (MultisP a)
```

Each stream function that can operate on Multi values consumes the Multis a in the sMultis field of the stream bundle. It must be prepared to accept either a MultisC or a MultisP, which is a "mixed" stream of scalars and Multi's. However, we always try to produce a MultisC and only fall back to a MultisP as a last resort. Even operations that can work with either representation are often worth specializing for the MultisC form. In the case of msum$_s$ above, this allows us to gobble up as many Multi values as possible and only then switch to consuming scalars, thereby cutting the number of accumulating parameters in half and reducing register pressure.

One could imagine attempting a representation that somehow guarantees longer runs of Multis, but this would add complexity and we doubt it would have any advantage over the MultisC representation, which has a distinct "phase shift" between operations on Multi and operations on scalars. For operations like zip that operate on multiple streams, we would need to guarantee that *both* streams have the same structure—it simply does not do to have one stream in the pair yield a scalar while the other yields a Multi. The MultiC/MultiP distinction neatly captures this requirement by framing it in terms of who has control over what is yielded next, consumers or producers.

3.2. A SIMD version of dotp

With a stream representation for SIMD computation in hand, we can write a SIMD-ized version of the dot product from Section 2:

```
dotp_simd :: Vector Double → Vector Double → Double
dotp_simd v w = msum (mzipWith (∗) v w)
```

The only difference with respect to the scalar implementation in Section 2.1 is that we use variants of foldl′ and zipWith specialized to take function arguments that operate on values that are members of the Num type class. While we could have used versions of these functions that take two function arguments (our library supports both options), one for scalars and one for Multis, the forms that use overloading to allow the function argument to be used at both the type a → a → a and Multi a → Multi a → Multi a are a convenient shorthand:

```
mfold′ :: (PackedVector Vector a, Num a, Num (Multi a))
       ⇒(∀b.Num b ⇒ b → b → b)
       → a → Vector a → a
mzipWith :: (PackedVector Vector a, Num a, Num
         (Multi a))
             ⇒ (∀b.Num b ⇒ b → b → b)
             → Vector a → Vector a → Vector a
msum :: (PackedVector Vector a, Num a, Num (Multi a))
     ⇒ Vector a → a
msum = mfold′ (+) 0
```

The particular fold we use here, mfold′, maintains two accumulators (a scalar and a Multi) when given a MultisP a and one accumulator when given a MultisC a. The initial value of the scalar accumulator is the third argument to mfold′ and the initial value of the Multi accumulator is formed by replicating this scalar argument across a Multi. The result of the fold is computed by combining the elements of the Multi accumulator and the scalar accumulator using the function multifold from Figure 1. Note that the first argument to mfold′ must be associative and commutative. The PackedVector type class constraint ensures both that the type a is an instance of MultiType and that elements contained in the vector are contiguous so that they can be extracted a Multi a at a time.

The stream version of mfold′, mfold′$_s$, can generate efficient code no matter what representation is contained in a Multis a. On the other hand, the stream version of mzipWith, mzipWith$_s$, requires that both its vector arguments have a MultisC representation. Since there is no good way to zip two streams when one yields a scalar and the other a Multi, if either bundle argument to mzipWith$_s$ does not have a MultisC representation available, mzipWith$_s$ falls back to an implementation that uses only scalar operations.

3.3. Automatic parallelization

Using SIMD instructions does not come entirely for free. Consider mapping over a vector represented using multis:

```
mmap :: (PackedVector Vector a)
     ⇒ (a → a)
     → (Multi a → Multi a)
     → Vector a → Vector a
```

To map efficiently over the vector, it does not suffice to pass a function of type (a → a), because that does not work over multis. We must also pass a semantically equivalent multi-version of the function. For simple arithmetic, matters are not too bad:

```
foo :: Vector Float → Vector Float
foo v = mmap (λx y → x + y ∗ 2) (λx y → x + y ∗ 2) v
```

The two lambdas are at different types, but Haskell's overloading takes care of that. We could attempt to abstract this pattern like this:

```
mmap :: (PackedVector Vector a)
     ⇒ (∀a.Num a ⇒ a → a)
     → Vector a → Vector a
```

But that attempt fails if you want operations in class Floating, say, rather than Num. What we want is a way to *automatically multi-ize scalar functions* (such as (λx y → x + y ∗ 2) above), so that we get a pair of a scalar function and a multi function, which in turn can be passed to map.

The programmer has to use mmap, which is a bit inconvenient. However, in separate work,[2, 13] the Data Parallel Haskell project has shown how to automatically vectorize programs; the target there was turning nested data parallelism into flat data parallelism, but it turns out that we can use the same technology to turn element-wise data parallelism into SIMD multi-style data parallelism. Putting together DPH and the ideas of this paper gives the best of both worlds: programmers can write data parallel programs without considering SIMD, and the compiler will automatically exploit the vector instructions if they are present. Better still, DPH allows us to take advantage of *multiple cores* as well as the SIMD units in each core.

We updated DPH to use our modified `vector` library. Because DPH programs are vectorized by the compiler so that all scalar operations are turned into operations over wide vectors, by implementing these wide vector operations using our new SIMD functions like msum, programs written using DPH automatically and transparently take advantage of SSE instructions—no code changes are required of the programmer. The full version of the paper includes benchmarks for our modified implementation of DPH.

3.4. How general is generalized stream fusion?

We do not mean to suggest that the representations we have chosen for our Bundle data type are complete in any sense except that they allow us to take advantage of bulk memory operations and SIMD instructions, which was our original goal. Generalized stream fusion is not "general" because we have finally hit upon the full set of representations one could possibly ever need, but because the frameworks we have put forth admit multiple new, specialized representations. The key features of generalized stream fusion are (1) the ability to add new specialized stream representations, notably without requiring the library writer to rewrite the entire library; (2) leveraging the compiler to statically eliminate all intermediate Bundle structures and leave behind the single representation that is actually necessary to perform the desired computation; and (3) not requiring the end user to know about the details of Bundles, or even that they exist.

Generalized stream fusion provides a representation and algebraic laws for rewriting operations over this representation whose usefulness extends beyond Haskell. Although we have implemented generalized stream fusion as a library, it could also be incorporated into a compiler as an intermediate language. This was not necessary in our implementation because GHC's generic optimizer is powerful enough to eliminate all intermediate structures created by generalized stream fusion. In other words, GHC is such a good partial evaluator that we were able to build generalized stream fusion as a library rather than incorporating it into the compiler itself. Writing high-level code without paying an abstraction tax is desirable in any language, and compilers other than GHC could also avoid this tax by using the ideas we outline in this paper, although perhaps only by paying a substantial one-time implementation cost.

4. IMPLEMENTATION

There are three substantial components of our implementation. We first modified GHC itself to add support for SSE instructions. This required modifying GHC's register allocator to allow overlapping register classes, which was necessary to allow SSE vectors to be stored in registers. We then added support for fully unboxed primitive SIMD vector types and primitive operations over these types to GHC's dialect of Haskell. The STG and C-intermediate languages as well as GHC's LLVM code generator, were also extended to support compiling the new Haskell SIMD primitives. Boxed wrappers for the unboxed primitives and the MultiType type class and its associated Multi type complete the high-level support for working directly with basic SIMD data types. Because the SIMD support we added to GHC utilizes the LLVM back-end, it should be relatively straightforward to adapt our modifications for other CPU architectures, although at this time only x86-64 is supported.

Second, we implemented generalized stream fusion in a modified version of the `vector` library[10] for computing with efficient unboxed vectors in Haskell. We replaced the existing stream fusion implementation with an implementation that uses the Bundle representation and extended the existing API with functions such as mfold′ and mzipWith that enable using SIMD operations on the contents of vectors. The examples in this paper are somewhat simplified from the actual implementations. For example, the actual implementations are written in monadic form and involve type class constraints that we have elided. Vectors whose scalar elements can be accessed in SIMD-sized groups, that is, vectors whose scalar elements are laid out consecutively in memory, are actually represented using a PackedVector type class. These details do not affect the essential design choices we have described, and the functions used in all examples are simply type-specialized instances of the true implementations.

Third, we modified the DPH libraries to take advantage of our new `vector` library. The DPH libraries are built on top of the stream representation from a previous version of the `vector` library, so we first updated DPH to use our bundle representation instead. We next re-implemented the primitive wide-vector operations in DPH in terms of our new SIMD operations on bundles. While we only provided SIMD implementation for operations on double-precision floating point values, this part of the implementation was quite small, consisting of approximately 20 lines of code not counting `#ifdef`s. Further extending SIMD support in DPH will be easy now that it is based on bundles rather than streams.

Our support for SSE and AVX instructions is part of the standard GHC distribution, and our modifications to the `vector` and DPH libraries are available in a public git repository.

5. EVALUATION

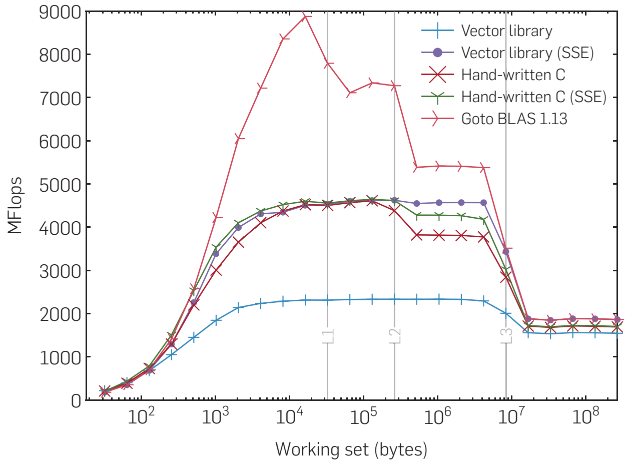
Our original goal in modifying GHC and the `vector` library was to make efficient use of SSE instructions from high-level Haskell code. The inability to use SSE operations from Haskell and its impact on performance is a deficiency that was brought to our attention by Lippmeier and Keller.[11] The first step we took was to write a small number of simple C functions utilizing SSE intrinsics to serve as benchmarks. This gave us a very concrete goal—to generate machine code from Haskell that was competitive with these C implementations. It is not a coincidence that one of the first such C functions that we wrote was an implementation of the vector dot product, in both a scalar version and a version using compiler intrinsics for manual SSE support. We omit the C versions, but repeat the definition of the Haskell implementation here:

```
ddotp :: Vector Double → Vector Double → Double
ddotp v w = mfold′ (+) 0 (mzipWith (∗) v w)
```

Though not exactly onerous, the C version with SSE support is already unpleasantly more complex than the scalar version. The Haskell version, consisting of a single line of code (not including the optional type signature), is certainly the simplest. Also note that the Haskell programmer can think compositionally—it is natural to think of dot product as pairwise multiplication followed by summation. The C programmer, on the other hand, must manually fuse the two loops into a single multiply-add. Furthermore, as well as being constructed compositionally, the Haskell implementation can itself be *used* compositionally. That is, if the input vectors to ddotp are themselves the results of vector computations, generalized stream fusion will potentially fuse *all* operations in the chain—not just the dot product's zip and fold—into a single loop. In contrast, the C programmer must manifest the input to the C implementation of `ddotp` as concrete vectors in memory—there is no potential for automatic fusion with other operations in the C version.

Figure 2 compares the single-threaded performance of several implementations of the dot product, including C and Haskell versions that only use scalar operations as well as the implementation provided by GotoBLAS2 1.13.[5,6] Times were measured on a 3.40 GHz Intel i7-2600K processor, averaged over 100 runs. To make the relative performance of the various implementations clearer, we show the execution time of each implementation relative to the scalar C version, which is normalized to 1.0, in Figure 3.

Figure 2. Single-threaded performance of double-precision dot product implementations. C implementations were compiled using GCC 4.8.1 and compiler options `-O3 -msse4.2 -ffast-math -ftree-vectorize -funroll-loops`. Sizes of the L1, L2, and L3 caches are marked.



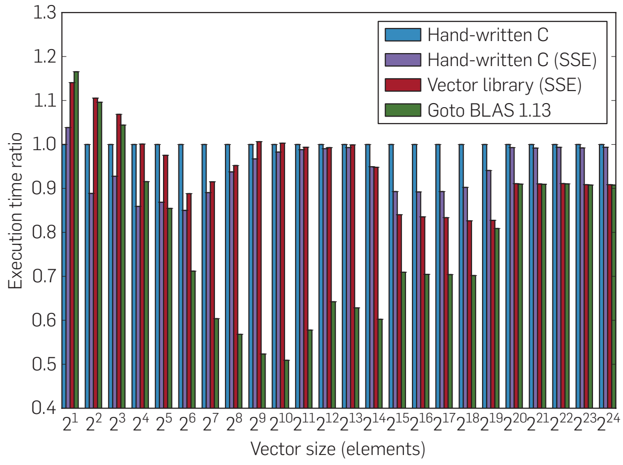
Surprisingly, both the naive scalar C implementation and the version written using SSE intrinsics perform approximately the same. This is because GCC automatically vectorizes the scalar implementation. However, the Haskell implementation is almost always faster than both C versions; it is 5–20% slower for very short vectors (those with fewer than about 16 elements) and 1–2% slower just at the point where the working set size exceeds the capacity of the L1 cache. Not only does Haskell outperform C on this benchmark, but it outperforms GCC's vectorizer. Once the working set no longer fits in L3 cache, the Haskell implementation is even neck-and-neck with the implementation of `ddotp` from GotoBLAS, a collection of highly tuned BLAS routines hand-written in assembly language that is generally considered to be one of the fastest BLAS implementation available.

5.1. Prefetching and loop unrolling

Why is Haskell so fast? Because in addition to leveraging loop fusion and a careful choice of representation, we have also exploited the high-level stream-fusion framework to embody two additional optimizations: *loop unrolling* and *prefetching*.

The generalized stream fusion framework allowed us to implement the equivalent of loop unrolling by adding under 200 lines of code to the `vector` library. We changed the MultisC data type to incorporate a *leap*, which is a Step that contains multiple values of type Multi a. We chose Leap to contain four values—so loops are unrolled four times—since on x86-64 processors this tends not to put too much

Figure 3. Relative performance of single-threaded `ddot` implementations. All times are normalized relative to the handwritten, compiler-vectorized, and C implementation.



register pressure on the register allocator. Adding multiple Leaps of different sizes would also be possible. MultisC consumers may choose not to use the Leap stepping function, in which case loops will not be unrolled:

```
data Leap a = Leap a a a a
data MultisC a where
  MultisC :: (s → Step s (Leap (Multi a)))
            → (s → Step s (Multi a))
            → (s → Step s a)
            → s
            → MultisC a
```

Prefetch instructions on Intel processors allow the program to give the CPU a hint about memory access patterns, telling it to prefetch memory that the program plans to use in the future. In our library, these prefetch hints are implemented using prefetch primitives that we added to GHC. When converting a Vector to a MultisC, we know exactly what memory access pattern will be used—each element of the vector will be accessed in linear order. The function that performs this conversion, stream, takes advantage of this knowledge by executing prefetch instructions as it yields each Leap. Only consumers using Leaps will compile to loops containing prefetch instructions, and stream will only add prefetch instructions for vectors whose size is above a fixed threshold (currently 8192 elements), because for shorter vectors the extra instruction dispatch overhead is not amortized by the increase in memory throughput. A prefetch distance of 128 * 12, based on the line fill buffer size of 128 bytes, was chosen empirically. Loop unrolling and prefetching produce an inner loop for our Haskell implementation of ddotp that is shown in Figure 4.[a]

Figure 4. Inner loop of Haskell `ddotp` function.

```
.LBB4_12:
        prefetcht0 1600(%rsi,%rdx)Z
        movupd     64(%rsi,%rdx), %xmm3
        prefetcht0 1600(%rdi,%rdx)
        movupd     80(%rsi,%rdx), %xmm0
        movupd     80(%rdi,%rdx), %xmm2
        mulpd      %xmm0, %xmm2
        movupd     64(%rdi,%rdx), %xmm0
        mulpd      %xmm3, %xmm0
        addpd      %xmm1, %xmm0
        addpd      %xmm2, %xmm0
        movupd     96(%rsi,%rdx), %xmm3
        movupd     96(%rdi,%rdx), %xmm1
        movupd     112(%rsi,%rdx), %xmm4
        movupd     112(%rdi,%rdx), %xmm2
        mulpd      %xmm4, %xmm2
        mulpd      %xmm3, %xmm1
        addq       $64, %rdx
        leaq       8(%rax), %rcx
        addq       $16, %rax
        addpd      %xmm0, %xmm1
        cmpq       %r9, %rax
        addpd      %xmm2, %xmm1
        movq       %rcx, %rax
        jle        .LBB4_12
```

Not only can the client of our modified `vector` library write programs in terms of boxed values and directly compose vector operations instead of manually fusing operations without paying an abstraction penalty, but he or she can transparently benefit from low-level prefetch "magic" baked into the library. Of course the same prefetch magic could be expressed manually in the C version. However, when we originally wrote the C implementation of dot product using SSE intrinsics, we did not know about prefetching. We suspect that many C programmers are in the same state of ignorance. In Haskell, this knowledge is embedded in a library, and clients benefit from it automatically.

6. RELATED WORK

Wadler[16] introduced the problem of deforestation, that is, of eliminating intermediate structures in programs written as compositions of list transforming functions. A great deal of follow-on work[4, 7, 8, 12, 14, 15] attempted to improve the ability of compilers to automate deforestation through program transformations. Each of these approaches to fusion has severe limitations. For example, Gill et al.[4] cannot fuse left folds, such as that which arises in sum, or zipWith, and Takano and Meijer[14] cannot handle nested computations such as mapping a function over concatenated lists. Our work is based on the stream fusion framework described by Coutts et al.,[3] which can fuse all of these use cases and more. The `vector` library uses stream fusion to fuse operations on vectors rather than lists, but the principles are the same.

7. CONCLUSION

Generalized stream fusion is a strict improvement on stream fusion; by re-casting stream fusion to operate on bundles of streams, each vector operation or class of operations can utilize a stream representation tailored to its particular pattern of computation. Though we focused on leveraging SSE instructions in this article, our implementation also adds support for efficient use of bulk memory operations in vector operations. As part of our work, we added support for low-level SSE instructions to GHC and incorporated generalized stream fusion into the `vector` library. Using our modified library, programmers can write compositional, high-level programs for manipulating vectors without loss of efficiency. Benchmarks show that these programs can perform competitively with hand-written C.

Although we implemented generalized stream fusion in a Haskell library, the bundled stream representation could be used as an intermediate language in another compiler. Vector operations would no longer be first class in such a formulation, but it would allow a language to take advantage of fusion without requiring implementations of the general purpose optimizations present in GHC that allow it to eliminate the intermediate structures produced by generalized stream fusion.

[a] The prefetch constant 1600 in the listing is 128 * 12 + 64 since the loop index in the generated assembly is offset by –64 bytes.

References

1. Chakravarty, M.M.T., Keller, G., Peyton Jones, S., Marlow, S. Associated types with class. In *Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL, 05 (New York, NY, USA, 2005). ACM, New York, NY, USA, 1–13.
2. Chakravarty, M.M.T., Leshchinskiy, R., Peyton Jones, S., Keller, G., Marlow, S. Data parallel Haskell: A status report. In *Proceedings of the 2007 Workshop on Declarative Aspects of Multicore Programming*, DAMP, 07

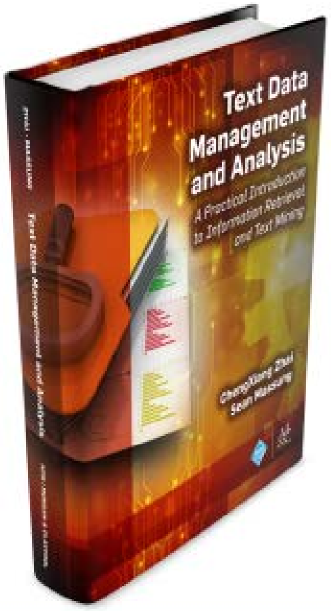
(Nice, France, 2007). ACM, New York, NY, USA, 10–18.

3. Coutts, D., Leshchinskiy, R., Stewart, D. Stream fusion: From lists to streams to nothing at all. In *Proceedings of the 12th ACM SIGPLAN International Conference on Functional Programming* (Freiburg, Germany, 2007). ACM, New York, NY, USA, 315–326.
4. Gill, A., Launchbury, J., Peyton Jones, S.L. A short cut to deforestation. In *Proceedings of the Conference on Functional Programming Languages and Computer Architecture*, FPCA, 93 (1993). ACM, New York, NY, USA, 223–232.
5. Goto, K., van de Geijn, R.A. Anatomy of high-performance matrix multiplication. *ACM Trans. Math. Softw. 34*, 3 (2008), 1–25.
6. Goto, K., van de Geijn, R. High-performance implementation of the Level-3 BLAS. *ACM Trans. Mathem. Softw. 35*, 1 (2008), 1–14.
7. Hamilton, G.W. Extending higher-order deforestation: Transforming programs to eliminate even more trees. In *Proceedings of the Third Scottish Functional Programming Workshop*, Hammond, K. and Curtis, S., eds. (Exeter, UK Aug. 2001). Intellect Books, 25–36.
8. Johann, P. Short cut fusion: Proved and improved. In *Proceedings of the 2nd International Workshop on Semantics, Applications, and Implementation of Program Generation*. Volume 2196 of *Lecture Notes in Computer Science* (Florence, Italy, 2001), 47–71.
9. Launchbury, J., Peyton Jones, S.L. State in Haskell. *Lisp Symb. Comput.* 8, 4 (1995), 293–341.
10. Leshchinskiy, R. Vector: Efficient arrays, Oct 2012. http://hackage.haskell.org/package/vector.
11. Lippmeier, B., Keller, G. Efficient parallel stencil convolution in Haskell. In *Proceedings of the 4th ACM Symposium on Haskell*, Haskell, 11 (2011). ACM, New York, NY, USA, 59–70.
12. Marlow, S., Wadler, P. Deforestation for higher-order functions. In *Proceedings of the 1992 Glasgow Workshop on Functional Programming* J. Launchbury and P. Sansom, eds. Workshops in Computing (1993). Springer-Verlag, London, UK, 154–165.
13. Peyton Jones, S., Leshchinskiy, R., Keller, G., Chakravarty, M.M.T. Harnessing the multicores: Nested data parallelism in Haskell. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, Hariharan, R., Mukund, M., and Vinay, V., eds. Volume 2 of *Leibniz International Proceedings in Informatics (LIPIcs)* (Dagstuhl, Germany, 2008) Schloss Dagstuhl -Leibniz-Zentrum fuer Informatik, 383–414.
14. Svenningsson, J. Shortcut fusion for accumulating parameters & zip-like functions. In *Proceedings of the Seventh ACM SIGPLAN International Conference on Functional Programming*, ICFP, 02 (Pittsburgh, PA, 2002). ACM, New York, NY, USA, 124–132.
15. Takano, A., Meijer, E. Shortcut deforestation in calculational form. In *Proceedings of the Seventh International Conference on Functional Programming and Computer Architecture*, FPCA, 95 (1995). ACM, New York, NY, USA, 306–313.
16. Wadler, P. Deforestation: Transforming programs to eliminate trees. *Theor. Comput. Sci. 73*, 2 (1990), 231–248.

Geoffrey Mainland (mainland@drexel.edu) Department of Computer Science, Drexel University, Philadelphia, PA.

Roman Leshchinskiy (rleshchinskiy@gmail.com).

Simon Peyton Jones (simonpj@microsoft.com) Microsoft Research Ltd Cambridge, England.


Text Data Management and Analysis
A Practical Introduction to Information Retrieval and Text Mining
ChengXiang Zhai
Sean Massung

DOI:10.1145/3060584

To view the accompanying paper, visit doi.acm.org/10.1145/3060586 rh

Technical Perspective
Building Knowledge Bases from Messy Data

By Alon Halevy

IMAGINE THE TASK of creating a database of all the high-quality specialty cafés around the world so you never have to settle for an imperfect brew. Relying on reviews from sites such as Yelp will not do the job because there is no restriction on who can post reviews there. You, on the other hand, are interested only in cafés that are reviewed by the coffee intelligentsia. There are several online sources with content relevant to your envisioned database. Cafés may be featured in well-respected coffee publications such as sprudge.com or baristamagazine.com, and data of more fleeting nature may pop up on your social media stream from coffee-savvy friends.

The task of creating such a database is surprisingly difficult. You would begin by deciding which attributes of cafés the database should model. Attributes such as address and opening hours would be obvious even to a novice, but you will need to consult a coffee expert who will suggest more refined attributes such as roast profile and brewing methods. The next step is to write programs that will extract structured data from these heterogeneous sources, distinguish the good extractions from the bad ones, and combine extractions from different sources to create tuples in your database. As part of the data cleaning process, you might want to employ crowd workers to confirm details, such as opening hours that were extracted from text or whether two mentions of cafés in text refer to the same café in the real world. In the extreme case, you might even want to send someone out to a café to check on some of the details in person. The process of creating the database is iterative because your extraction techniques will be refined and because the café scene changes frequently.

This Knowledge Base Construction task (KBC) has been an ongoing challenge and an inspiration for deep collaborations between researchers and practitioners in multiple fields, including data management and integration, information extraction, machine learning, natural language understanding, and probabilistic reasoning. Aside from the compelling application detailed here, the problem arises in many other settings where we need to construct databases from messy data. For example, imagine the task of creating a database (or ontology) of all job categories for a job-search site, or compiling a database of dishes served in Tokyo restaurants for the purpose of restaurant search or trend analysis.

The following paper is a prime example of groundbreaking work in the area of KBC. DeepDive, a project led by Chris Ré at Stanford, is an end-to-end system for creating knowledge bases. The input to DeepDive is a set of data sources such as text documents, PDF files, and structured databases. DeepDive extracts, cleans, and integrates data from the multiple sources and produces a database in which a probability is attached to every tuple. A user interacts with DeepDive in a high-level declarative language (DDLog) that uses predicates defined with functions in Python. The rules in DDLog specify how to extract entities, mentions of entities, and relationships from the data sources and the details of the extractions are implemented in Python. Based on this specification, DeepDive then uses an efficient statistical inference engine to compute probabilities of the facts in the database. Using a set of tools that facilitate examining erroneous extractions, the user can iteratively adjust the DDLog rules to obtain the desired precision and recall. DeepDive has already been used in several substantial applications, such as detecting human trafficking and creating a knowledge base for paleobiologists with quality higher than human volunteers.

The following paper is a prime example of groundbreaking work in the area of Knowledge Base Construction.

One of the areas the DeepDive project focused on in particular is the incremental aspect of building a database. As noted, in several applications of the system, knowledge base construction is an iterative process. As the user goes through the process of building the knowledge base, the rules used to extract the data change and, of course, the underlying data may change as well. The DeepDive project developed algorithms to efficiently recompute the facts in the knowledge base and to efficiently recompute the probabilities of facts coming from the inference engine. The results show that efficient incremental computation can make a substantial difference in the usability of a KBC system.

Like with any deep scientific endeavor, there is much more research to be done (and for now, too many coffee lovers need to settle for over-roasted coffee because the database of cafés does not exist yet). We hope that reading this paper will inspire you to work on the KBC problem and hopefully to contribute ideas from far-flung fields. C

Alon Halevy is CEO of the Recruit Institute of Technology (R.I.T), Mountain View, CA.

DOI:10.1145/3060586

DeepDive: Declarative Knowledge Base Construction

By Ce Zhang, Christopher Ré, Michael Cafarella, Christopher De Sa, Alex Ratner, Jaeho Shin, Feiran Wang, and Sen Wu

Abstract

The dark data extraction or knowledge base construction (KBC) problem is to populate a relational database with information from unstructured data sources, such as emails, webpages, and PDFs. KBC is a long-standing problem in industry and research that encompasses problems of data extraction, cleaning, and integration. We describe DeepDive, a system that combines database and machine learning ideas to help to develop KBC systems. The key idea in DeepDive is to frame traditional extract–transform–load (ETL) style data management problems as a single large statistical inference task that is declaratively defined by the user. DeepDive leverages the effectiveness and efficiency of statistical inference and machine learning for difficult extraction tasks, whereas not requiring users to directly write any probabilistic inference algorithms. Instead, domain experts interact with DeepDive by defining features or rules about the domain. DeepDive has been successfully applied to domains such as pharmacogenomics, paleobiology, and antihuman trafficking enforcement, achieving human-caliber quality at machine-caliber scale. We present the applications, abstractions, and techniques used in DeepDive to accelerate the construction of such dark data extraction systems.

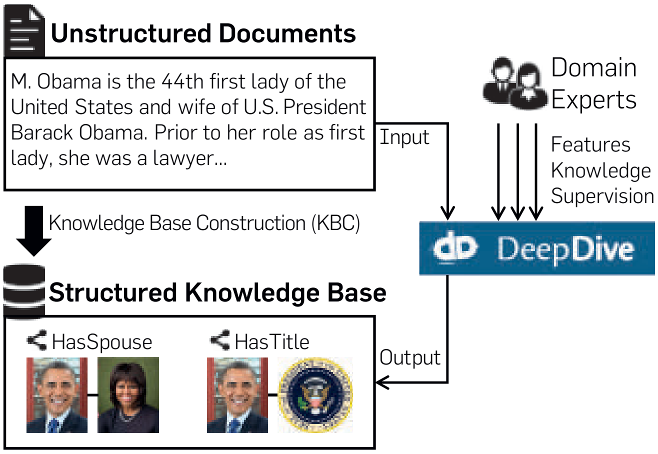
1. INTRODUCTION

The goal of knowledge base construction (KBC) is to populate a structured relational database from unstructured input sources, such as text documents, PDFs, and diagrams. As the amount of available unstructured information has skyrocketed, this task has become a critical component in enabling a wide range of new analysis tasks. For example, analyses of protein–protein interactions for biological, clinical, and pharmacological applications[29]; online human trafficking activities for law enforcement support; and paleological facts for macroscopic climate studies[36] are all predicated on leveraging data from large volumes of text documents. This data must be collected in a structured format in order to be used, however, and in most cases doing this extraction by hand is untenable, especially when domain expertise is required. Building an automated KBC system is thus often the key development step in enabling these analysis pipelines.

The process of populating a structured relational database from unstructured sources has also received renewed interest in the database community through high-profile start-up companies, established companies such as IBM's Watson,[5, 15] and a variety of research efforts.[9, 26, 31, 41, 46] At the same time, the natural language processing and machine learning communities are attacking similar problems.[3, 12, 22] Although different communities place differing emphasis on the extraction, cleaning, and integration phases, all seem to be converging toward a common set of techniques that includes a mix of data processing, machine learning, and engineers-in-the-loop.[a]

Here, we discuss DeepDive, our open-source engine for constructing knowledge bases with human-caliber quality at machine-caliber scale (Figure 1). DeepDive takes the viewpoint that in information extraction, the problems of extraction, cleaning, and integration are not disjoint algorithmic problems, though the database community has treated them as such for several decades. Instead, these problems can be more effectively attacked jointly, and viewed as a single statistical inference problem that takes all available information into account to produce the best possible end result. We have found that one of the most harmful inefficiencies of traditional pipelined approaches is that developers struggle to understand how changes to the separate extraction, cleaning, or integration modules improve the overall system quality, leading them to incorrectly distribute their development

Figure 1. Knowledge base construction (KBC) is the process of populating a structured relational knowledge base from unstructured sources. DeepDive is a system aimed at facilitating the KBC process by allowing domain experts to integrate their domain knowledge without worrying about algorithms.


Unstructured Documents
M. Obama is the 44th first lady of the United States and wife of U.S. President Barack Obama. Prior to her role as first lady, she was a lawyer...
Input
Domain Experts
Features
Knowledge
Supervision
Knowledge Base Construction (KBC)
DeepDive
Structured Knowledge Base
HasSpouse
HasTitle
Output

[a] http://deepdive.stanford.edu.

The original version of this paper is entitled "Incremental Knowledge Base Construction Using DeepDive" and was published in *Proceedings of the VLDB Endowment*, 2015. This paper also contains content from other previously published work.[16, 36, 39, 51] efforts. For example, developers might decide to sink a large amount of time into improving the quality of some upstream component of their pipeline, only to find that it has a negligible effect on end system performance—essentially, running into an Amdahl's law for quality. In contrast, by formulating the task as a single probabilistic inference problem, DeepDive allows the developer to effectively profile the end-to-end quality of his or her application. We argue that our approach leads to higher quality end-to-end models in less time, which is the ultimate goal of all information extraction systems.

Like other KBC systems, DeepDive uses a high-level declarative language to enable the user to describe application inputs, outputs, and model structure.[9, 31, 33] DeepDive's language is based on SQL, but also inherits Markov Logic Networks' formal semantics to enable users to declaratively describe their KBC task as a type of probabilistic graphical model called a *factor graph*.[11, 33]

DeepDive uses a standard execution model[9, 31, 33] in which programs go through two main phases, grounding and inference. In the *grounding* phase, DeepDive evaluates a sequence of SQL queries to produce a *factor graph* that describes a set of random variables and how they are correlated. Essentially, every tuple in the database which represents a *candidate* extraction to be potentially included in the output knowledge base is included as a random variable (node) in this factor graph. In the *inference* phase, DeepDive then takes the factor graph from the grounding phase and performs statistical inference using standard techniques, for example, Gibbs sampling.[47, 50] The output of inference is the marginal probability of every tuple in the output knowledge base. As with Google's Knowledge Vault[12] and others,[34] DeepDive also produces marginal probabilities that are *calibrated*: if one examined all facts with probability 0.9, we would expect approximately 90% of these facts to be correct. To calibrate these probabilities, DeepDive estimates (i.e., learns) parameters of the statistical model from data. Inference is a subroutine of the learning procedure and is the critical loop. Inference and learning are computationally intense (hours on 1TB RAM/48-core machines).

In our experience, we have found that DeepDive can reliably obtain extremely high quality on a range of KBC tasks. In the past few years, DeepDive has been used to build dozens of high-quality KBC systems by a handful of technology companies, a number of law enforcement agencies via DARPA's MEMEX program, and scientists in fields, such as paleobiology, drug repurposing, and genomics. Recently, we compared the quality of a DeepDive system's extractions to those provided by human volunteers over the last 10 years for a paleobiology database, and we found that the DeepDive system had higher quality (both precision and recall) on many entities and relationships. Moreover, on all of the extracted entities and relationships, DeepDive had no worse quality.[36] Additionally, the winning entry of the 2014 TAC-KBC competition was built on DeepDive.[1]

One key lesson learned was that in all cases, enabling developers to iterate quickly was critical to achieving such high quality. More broadly, we have seen that the process of developing KBC systems for real applications is fundamentally iterative: quality requirements change, new data sources arrive, and new concepts are needed in the application. Thus, DeepDive's architecture is designed around a set of techniques that not only make the execution of statistical inference and learning efficient, but also make the entire pipeline incremental in the face of changes both to the data and to the declarative specification.

This article aims at giving a broad overview of DeepDive. The rest of the article is organized as follows. Section 2 describes some example applications of DeepDive and outlines core technical challenges. Section 3 presents the system design and language for modeling KBC systems inside DeepDive. We discuss the different techniques in Section 4 and give pointers for readers who are interested in each technique.

2. APPLICATIONS AND CHALLENGES

KBC plays a critical role in many analysis tasks, both scientific and industrial, and is often *the* bottleneck to answering new and impactful macroscopic questions. In many scientific analyses, for example, one first needs to assemble a large, high-quality knowledge base of facts (typically from the literature) in order to understand macroscopic trends and patterns, for example, about the amount of carbon in the Earth's atmosphere throughout time[36] or all the drugs that interact with a particular gene,[29] and some scientific disciplines have undertaken decade-long collection efforts to this end, for example, PaleoDB.org and PharmaGKB.org.

In parallel, KBC has attracted interest from industry[15, 52] and many areas of academia outside of computer science.[2, 3, 6, 14, 23, 25, 31, 34, 37, 41, 43, 48] To understand the common patterns in KBC systems, we are actively collaborating with scientists from a diverse set of domains, including geology,[49] paleontology,[36] pharmacology for drug repurposing, and others. We first describe one KBC application we built, called PaleoDeepDive, then present a brief description of other applications built with similar purposes and finally discuss the challenges inherent in building such systems.

2.1. PaleoDB and PaleoDeepDive

Paleontology is based on the description and biological classification of fossils, an enterprise that has been recorded in hundreds to thousands of scientific publications over the past four centuries. One central task that paleontologists have long been concerned with is the construction of a knowledge base about fossils from scientific publications. Existing knowledge bases compiled by human volunteers—for example, PaleoDB—have already greatly expanded the intellectual reach of paleontology and led to many fundamental new insights into macroevolutionary processes and the nature of biotic responses to global environmental change. However, the current process of using human volunteers is usually expensive and time-consuming. For example, PaleoDB, one of the largest such knowledge bases, took more than 300 professional paleontologists and 11 human years to build over the last two decades, resulting in `PaleoDB.org`. To get a sense of the impact of this database on this field, at the time of writing, this dataset has contributed to 205 publications, of which 17 have appeared in *Nature* or *Science*.

The potential impact of automating this labor-intensive extraction task and the difficulty of the task itself provided an ideal test bed for our KBC research. In particular, we constructed a prototype called PaleoDeepDive[36] that takes in PDF documents and extracts a set of paleontological entities and relations (see Figure 2). This prototype attacks challenges in optical character recognition, natural language processing, information extraction, and integration. Some statistics about the process are shown in Figure 3. As part of the validation of this system, we performed a double-blind experiment to assess the quality of PaleoDeepDive versus PaleoDB. We found that PaleoDeepDive achieved accuracy comparable to—and sometimes better than—that of PaleoDB (see Figure 3).[36] Moreover, PaleoDeepDive was able to process roughly 10x the number of documents, with per-document recall roughly 2.5x that of human annotators.

2.2. Beyond paleontology

The success of PaleoDeepDive motivates a series of other KBC applications in a diverse set of domains, including both natural and social sciences. Although these applications focus on very different types of KBs, they are usually built in a way similar to PaleoDeepDive. This similarity across applications has motivated us to build DeepDive as a unified framework to support these diverse applications.

Human trafficking. Human trafficking is an odious crime that uses physical, economic, or other means of coercion to obtain labor from human beings, who are often used in sex or factory work. Identifying victims of human trafficking is difficult for law enforcement using traditional means; however, like many other forms of commerce, sex work advertising is now online, where providers of sex services post ads containing price, location, contact information, physical characteristics, and other data. As part of the DARPA MEMEX project, we ran DeepDive on approximately 90M advertisements and 0.5M forum posts, creating two distinct structured tables that included extracted attributes about potentially trafficked workers, such as price, location, phone number, service types, age, and various other attributes that can be used to detect signs of potential trafficking or abuse. In many cases, DeepDive is able to extract these attributes with comparable or greater quality levels than human annotators; for example, on phone number extraction from service ads, DeepDive achieves 99.5% precision and 95.5% recall, whereas human annotators only obtain 93% precision and 92.5% recall. MEMEX has been covered on *60 min* and other news sources, currently supports the operations of several law enforcement agencies nationwide, and has been used in at least one arrest and conviction.

Medical genetics. The body of literature in the life sciences has been growing at an accelerating speed, to the extent that it has been unrealistic for scientists to perform research solely based on reading and/or keyword search. Numerous manually curated structured knowledge bases are likewise unable to keep pace with exponential increases in the number of publications available online. For example, OMIM is an authoritative database of human genes and Mendelian genetic disorders that dates back to the 1960s, and so far contains about 6000 hereditary diseases or phenotypes, growing at a rate of roughly 50 records per month for many years. Conversely, almost 10,000 publications were deposited into PubMed Central per month last year. In collaboration with Prof. Gill Bejerano at Stanford, we are developing DeepDive applications to create knowledge bases in the field of medical genetics. Specifically, we use DeepDive to extract mentions of direct causal relationships between specific gene variants and clinical phenotypes from the literature that are presently being applied to clinical genetic diagnostics and reproductive counseling.[b]

Pharmacogenomics. Understanding the interactions of chemicals in the body is a key to drug discovery. However, the majority of this data resides in the biomedical literature and cannot be easily accessed. The Pharmacogenomics Knowledge Base is a high quality database that aims to annotate the relationships between drugs, genes, diseases, genetic variation, and pathways in the literature. In collaboration with Emily Mallory and Prof. Russ Altman at Stanford, we used DeepDive to extract mentions of gene–gene interactions from the scientific literature,[29] and are currently developing DeepDive applications with extraction schemas that include relations between genes, diseases, and drugs in order to predict novel pharmacological relationships.[c]

TAC-KBP. TAC-KBP is a NIST-sponsored research competition in which the task is to extract common properties

[b] http://www.cbsnews.com/news/new-search-engine-exposes-the-dark-web/.

[c] https://www.pharmgkb.org/.

Figure 2. Example relations extracted from text, tables, and diagrams in the paleontology literature by PaleoDeepDive.

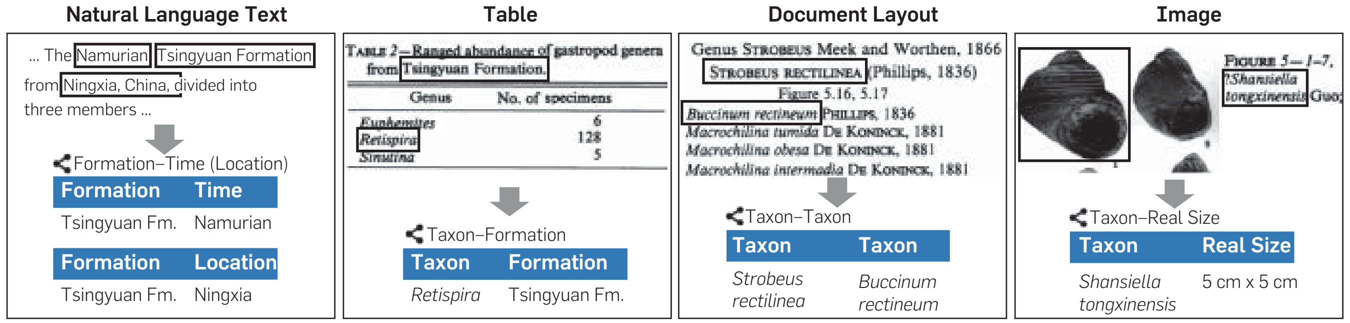

Natural Language Text
... The Namurian Tsingyuan Formation from Ningxia, China, divided into three members ...
Formation–Time (Location)
Formation | Time
Tsingyuan Fm. | Namurian
Formation | Location
Tsingyuan Fm. | Ningxia
Table
TABLE 2—Ranged abundance of gastropod genera from Tsingyuan Formation.
Genus | No. of specimens
Euphemites 6
Retispira 128
Sinuitina 5
Taxon–Formation
Taxon | Formation
Retispira | Tsingyuan Fm.
Document Layout
Genus STROBEUS Meek and Worthen, 1866
STROBEUS RECTILINEA (Phillips, 1836)
Figure 5.16, 5.17
Buccinum rectineum PHILLIPS, 1836
Macrochilina tumida DE KONINCK, 1881
Macrochilina obesa DE KONINCK, 1881
Macrochilina intermadia DE KONINCK, 1881
Taxon–Taxon
Taxon | Taxon
Strobeus rectilinea | Buccinum rectineum
Image
FIGURE 5—1–7, ?Shansiella tongxinensis Guo;
Taxon–Real Size
Taxon | Real Size
Shansiella tongxinensis | 5 cm x 5 cm

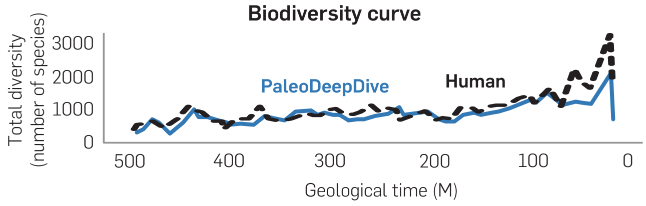
Figure 3. Quality of KBC systems built with DeepDive. On many applications, KBC systems built with DeepDive achieve comparable (and sometimes better) quality than professional human volunteers, and lead to similar scientific insights on topics, such as biodiversity. This quality is achieved by iteratively integrating diverse sources of data-often quality scales with the amount of information we enter into the system.

Quality of PaleoDeepDive

Relation	PDD # ext. (M)	PDD accuracy (%)	Human accuracy (%)
Taxon–Taxon	27	97	92
Taxon–Fm.	4	96	84
Fm.–Time	3	92	89
Fm.–Location	5	94	90

Scale of PaleoDeepDive

Metric	PDD	PDB
Documents processed	300K+	40K
Test set extractions	129K	60K



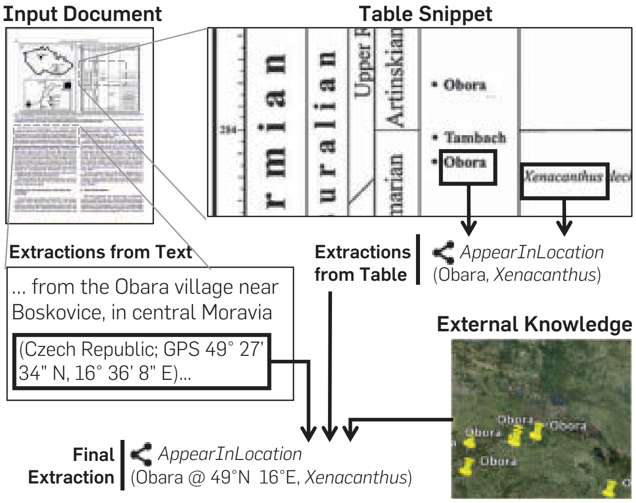
of people and organizations (e.g., age, birthplace, spouses, and shareholders) from 1.3 million newswire and web documents—this task is also termed *slot filling*. In the 2014 evaluation, 31 US and international teams participated in the competition, including a Stanford team that submitted a solution based on DeepDive.[1] The DeepDive-based solution achieved the highest precision, recall, and F1 of all the submissions.

2.3. Challenges

In all the applications mentioned above, KBC systems built with DeepDive achieved high quality as illustrated in Figure 3. Achieving this high quality level requires that we deal with several challenging aspects of the KBC problem.

Unstructured data complexity. In its full generality, the KBC task encompasses several longstanding grand challenges of computer science, including machine reading and computer vision. Even for simple schemas, extraction of structured information from unstructured sources contains many challenging aspects. For example, consider extracting the relation `Causes(Gene, Phenotype)`—that is, assertions of a genetic mutation causing a certain phenotype (symptom)—from the scientific literature (see Section 2.2). Genes generally have standardized forms of expression (e.g., BRCA1); however, they are easily confused with acronyms for diseases they cause; signals from across the document must be used to resolve these false positives. Phenotypes are even more challenging, because they can be expressed in many synonymous forms (e.g., "headache," "head pain," and "pain in forehead"). And extracting pairs that participate in the `Caused` relation encompasses dealing with all the standard challenges of linguistic variation and complexity, as well as application-specific domain terminology.

Figure 4. One challenge of building high-quality KBC systems is exploiting diverse sources of information jointly to extract data accurately. In this example page of a *Paleontology* journal article, identifying the correct location of *Xenacanthus* requires integrating information from within tables, text, and external structured knowledge bases. This problem becomes even more challenging when many extractors are not 100% accurate, motivating the joint probabilistic inference engine inside DeepDive.



This challenge becomes even more serious when information comes from different sources that potentially need to be considered together—that is, *jointly*—to make a correct extraction. In Figure 4, for example, to reach the extraction that the genus *Xenacanthus* appears in the location of the name *Obara*, the extraction system needs to consult extractions from text, tables, and external structured sources.

Scale. KBC systems need to be able to ingest massive numbers of documents, far outstripping the document counts of even well-funded human curation efforts. For example, Figure 5 illustrates the data flow of PaleoDeepDive. The input to PaleoDeepDive contains nearly 300K journal articles and books, whose total size exceeds 2TB. These raw inputs are then processed with tools such as OCR and linguistic parsing, which are computationally expensive and may take hundreds of thousands of machine hours.[d]

Multimodal input. We have found that text is often not enough: often, the data that are interesting to scientists are located in the tables, figures, and images of articles. For example, in geology, more than 50% of the facts that we are interested in are buried in tables.[16] For paleontology, the relationship between taxa, as known as taxonomy, is almost exclusively expressed in section headers.[36] For pharmacology, it is not uncommon for a simple diagram to contain a large number of metabolic pathways. Additionally, external sources of information (other knowledge bases) typically contain high-quality signals (e.g., Freebase and Macrostrat) that we would like to leverage and integrate. To build a high-quality KBC system, we need to deal with these diverse modalities of input.[e]

[d] http://www.freebase.com/.

[e] http://macrostrat.org/.

Figure 5. Another challenge of building high-quality KBC systems is that one usually needs to deal with data at the scale of terabytes. These data are not only processed with traditional relational operations, but also operations involving machine learning and statistical inference. Thus, DeepDive consists of a set of techniques to increase the speed, scale, and incremental execution of inference tasks involving billions of correlated random variables.

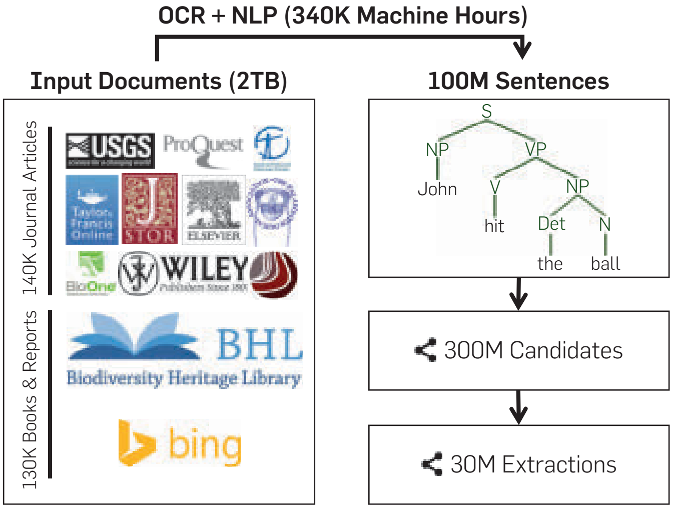
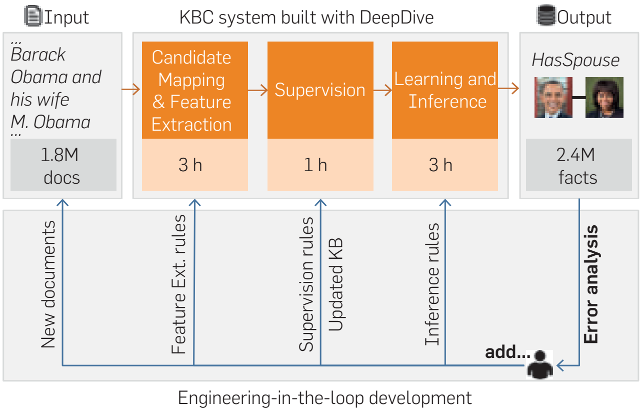


Figure 6. A KBC system takes unstructured documents as input and outputs a structured knowledge base. The runtimes are for the TAC-KBP competition system. To improve quality, the developer adds new rules and new data with error analysis conducted on the result of the current snapshot of the system. DeepDive provides a declarative language to specify each type of different rule and data, and techniques to incrementally execute this iterative process.



3. KBC USING DEEPDIVE

We describe DeepDive, an end-to-end framework for building KBC systems with a declarative language.

3.1. Definitions for KBC systems

The *input* to a KBC system is a heterogeneous collection of unstructured, semistructured, and/or structured data, ranging from text documents to existing but incomplete KBs, and an *application schema* specifying the target relations to extract. The *output* of the system is a relational database containing relations extracted from the input according to the application schema. Creating the knowledge base involves extraction, cleaning, and integration.

Example 3.1. *Figure 6 illustrates a running example in which our goal is to construct a knowledge base with pairs of individuals who are married to each other. The input to the system is a collection of news articles and an incomplete set of married people; the output is a Knowledge base (KB) containing pairs of people that the input sources assert to be married. A KBC system extracts linguistic patterns, for example, "... and his wife ..." between a pair of mentions of individuals (e.g., Barack Obama and M. Obama); these patterns are then used as features in a classifier deciding whether this pair of mentions indicates that they are married (in the* HasSpouse*) relation.*

We adopt standard terminology from KBC, for example, ACE. There are four types of objects that a KBC system seeks to extract from input documents, namely *entities*, *relations*, *mentions*, and *relation mentions*. An *entity* is a real-world person, place, or thing. For example, "Michelle_Obama_1" represents the actual entity for a person whose name is "Michelle Obama"; another individual with the same name would have another number. A *relation* associates two (or more) entities, and represents the fact that there exists a relationship between the participating entities. For example, "Barack_Obama_1" and "Michelle_Obama_1" participate in the HasSpouse relation, which indicates that they are married. These real-world entities and relationships are described in text. A *mention* is a span of text in an input document that refers to an entity or relationship: "Michelle" may be a mention of the entity "Michelle_Obama_1." A *relation mention* is a phrase that connects two mentions that participate in a relation, such as Barack Obama and M. Obama. The process of mapping mentions to entities is called *entity linking*.

3.2. The DeepDive framework[f]

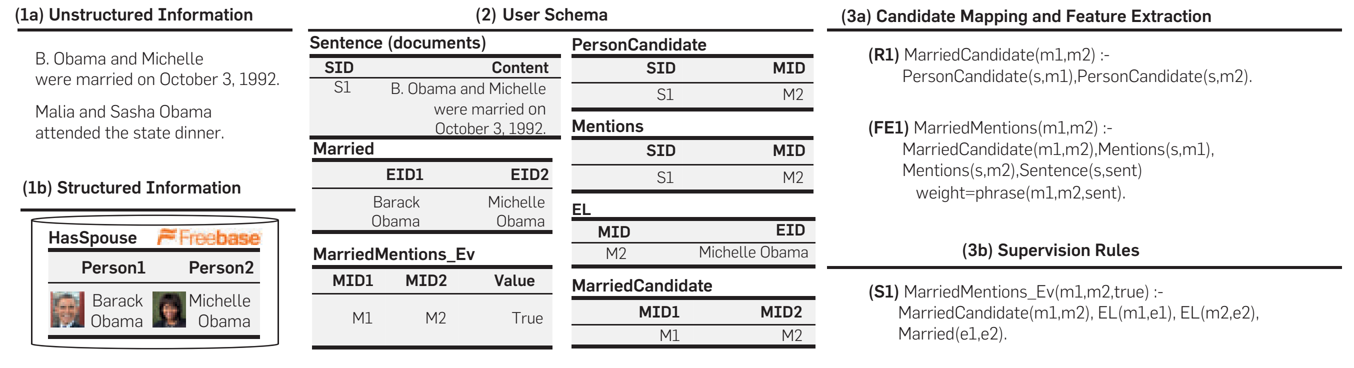
DeepDive is an end-to-end framework for building KBC systems. In this section, we walk through each phase. DeepDive supports both SQL and Datalog, but we use datalog syntax for this exposition. The rules we describe in this section are manually created by the user of DeepDive, and the process of creating these rules is application-specific. For simplicity of exposition, we focus on an example with text input in the rest of the section (Figure 7).[g]

Candidate mapping and feature extraction. All data in DeepDive—preprocessed input, intermediate data, and final output—is stored in a relational database. The first phase populates the database using a set of SQL queries and user-defined functions (UDFs) that we call *feature extractors*. By default, DeepDive stores all documents in the database in one sentence per row with markup produced by standard

[f] http://www.itl.nist.gov/iad/mig/tests/ace/2000/.

[g] For more information, including examples, please see http://deepdive.stanford.edu. Note that our engine is built on Postgres and Greenplum for all SQL processing and UDFs. There is also a port to MySQL.

Figure 7. An example KBC system (see Section 3.2 for details).



(1a) Unstructured Information

B. Obama and Michelle were married on October 3, 1992.

Malia and Sasha Obama attended the state dinner.

(1b) Structured Information

HasSpouse

Person1	Person2
Barack Obama	Michelle Obama

(2) User Schema

Sentence (documents)

SID	Content
S1	B. Obama and Michelle were married on October 3, 1992.

Married

EID1	EID2
Barack Obama	Michelle Obama

MarriedMentions_Ev

MID1	MID2	Value
M1	M2	True

PersonCandidate

SID	MID
S1	M2

Mentions

SID	MID
S1	M2

EL

MID	EID
M2	Michelle Obama

MarriedCandidate

MID1	MID2
M1	M2

(3a) Candidate Mapping and Feature Extraction

(R1) MarriedCandidate(m1,m2) :-
PersonCandidate(s,m1),PersonCandidate(s,m2).

(FE1) MarriedMentions(m1,m2) :-
MarriedCandidate(m1,m2),Mentions(s,m1),
Mentions(s,m2),Sentence(s,sent)
weight=phrase(m1,m2,sent).

(3b) Supervision Rules

(S1) MarriedMentions_Ev(m1,m2,true) :-
MarriedCandidate(m1,m2), EL(m1,e1), EL(m2,e2),
Married(e1,e2).

NLP preprocessing tools, including HTML stripping, part-of-speech tagging, and linguistic parsing. After this loading step, DeepDive executes two types of queries: (1) *candidate mappings*, which are SQL queries that produce possible mentions, entities, and relations, and (2) *feature extractors*, which associate features to candidates, for example, "… and his wife …" in Example 3.1.

Example 3.2. *Candidate mappings are usually simple. Here, we create a relation mention for every pair of candidate persons in the same sentence (s):*

(R1) `MarriedCandidate`(m1, m2): -
`PersonCandidate`(s,m1), `PersonCandidate`(s,m2).

Candidate mappings are simply SQL queries with UDFs that look like low-precision but high-recall extract–transform–load (ETL) scripts. Such rules must be high recall: if the union of candidate mappings misses a fact, DeepDive has no chance to extract it.

We also need to extract features, and we extend classical Markov logic[11] in two ways: (1) *user-defined functions (UDFs)* and (2) *weight tying*, which we illustrate by example.

Example 3.3. *Suppose that* phrase(m1, m2, sent) *returns the phrase between two mentions in the sentence, for example, "and his wife" in the above example. The phrase between two mentions may indicate whether two people are married. We would write this as:*

(FE1) `MarriedMentions`(m1, m2) : -
`MarriedCandidate`(m1, m2), `Mention`(s, m1),
`Mention`(s, m2), `Sentence`(s, sent)
weight = phrase(m1, m2, sent).

One can think about this as a classifier: This rule says that whether the text indicates that the mentions m1 *and* m2 *are married is influenced by the phrase between those mention pairs. The system will infer, based on training data, its confidence (by estimating the weight) that two mentions are indeed indicated to be married.*

Technically, phrase returns an identifier that determines which weights should be used for a given relation mention in a sentence. If phrase returns the same result for two relation mentions, they receive the *same* weight. We explain weight tying in more detail in Section 3.3. In general, phrase could be an arbitrary UDF that operates in a per-tuple fashion. This allows DeepDive to support common feature types ranging from "bag-of-words" to context-aware NLP features to feature sets incorporating domain-specific dictionaries and ontologies. In addition to specifying sets of classifiers, DeepDive inherits Markov Logic's ability to specify rich correlations between entities via weighted rules. Such rules are particularly helpful for data cleaning and data integration.

Supervision. Just as in Markov Logic, DeepDive can use training data or evidence about any relation; in particular, each user relation is associated with an evidence relation with the same schema and an additional field that indicates whether the entry is true or false. Continuing our example, the evidence relation MarriedMentions_Ev could contain mention pairs with positive and negative labels. Operationally, two standard techniques generate training data: (1) hand-labeling and (2) *distant supervision*, which we illustrate here.

Example 3.4. *Distant supervision*[19, 30] *is a popular technique to create evidence in KBC systems. The idea is to use an incomplete KB of married entity pairs to* heuristically *label (as* `True` *evidence) all relation mentions that link to a pair of married entities:*

(S1) `MarriedMentions_Ev`(m1, m2, true) : -
`MarriedCandidates`(m1, m2), `EL`(m1, e1),
`EL`(m2, e2), `Married`(e1, e2).

Here, `Married` *is an (incomplete) list of married real-world persons that we wish to extend. The relation EL is for "entity linking" that maps mentions to their candidate entities. At first blush, this rule seems incorrect. However, it generates noisy, imperfect examples of sentences that indicate two people are married. Machine learning techniques are able to exploit redundancy to cope with the noise and learn the relevant phrases (e.g., and his wife). Negative examples are generated by relations that are largely disjoint (e.g., siblings). Similar to DIPRE*[4] *and Hearst patterns,*[18] *distant supervision exploits the "duality"*[4] *between patterns and relation instances; furthermore, it allows us to integrate this idea into DeepDive's unified probabilistic framework.*

Learning and inference. In the learning and inference phase, DeepDive generates a factor graph, similar to Markov Logic, and uses techniques from Tuffy.[33] The inference and learning are done using standard techniques (Gibbs sampling) that we describe below after introducing the formal semantics.

Error analysis. DeepDive runs the above three phases in sequence, and at the end of the learning and inference, it obtains a marginal probability p for each candidate fact. To produce the final KB, the user selects facts that DeepDive predicts are true with probability above some user-selected threshold, for example, $p > 0.95$. Typically, the user needs to inspect errors and repeat the previous steps, a process that we call *error analysis*. Error analysis is the process of understanding the most common mistakes (incorrect extractions, overly specific features, candidate mistakes, etc.) and deciding how to correct them.[39] To facilitate error analysis, users write standard SQL queries.

3.3. Discussion of design choices

We have found the following key aspects of the DeepDive approach that we believe enable noncomputer scientists to build sophisticated KBC systems: (1) There is no reference in a DeepDive program to the underlying machine learning algorithms. Thus, DeepDive programs are declarative in a strong sense. Probabilistic semantics provide a way to debug the system independent of the algorithm it uses. (2) DeepDive allows users to write feature extraction code (UDFs) in familiar languages (Python, SQL, and Scala). (3) By using and producing relational databases, DeepDive fits into the familiar SQL stack, which allows standard tools to inspect and visualize the data. (4) The user constructs an end-to-end system and then refines the quality of the system in a pay-as-you-go way.[28] In contrast, traditional pipeline-based ETL scripts may lead to a user's time and effort being overspent on a specific extraction or integration step—without the ability to evaluate how important each step is for the quality of the end result. Anecdotally, pay-as-you-go leads to more informed decisions about how to improve quality.

The above design choices necessitated overcoming several technical challenges, two of which we briefly highlight below.

Joint statistical inference. In many systems, successive stages are simply pipelined together, propagating errors from one stage to the next, and complicating iterative development efforts. This can also have noticeable performance effects when the information from different sources are all *noisy*, and potentially need to be considered together—that is, *jointly*—to make a correct extraction. To join extractions with different confidence levels together, one needs a principled framework. The DeepDive approach to this challenge is based on a Bayesian probabilistic approach. DeepDive treats all these information sources as one joint probabilistic inference problem, with all predictions modeled as random variables within a factor graph model. This probabilistic framework ensures that all facts produced by DeepDive are associated with a marginal probability. These marginal probabilities are meaningful in DeepDive; that is, they represent the empirical accuracy that one should expect for the extracted mentions, and provide a guideline to the developer for improving the KBC system built using DeepDive.

Incremental and efficient execution. Especially with the above design choices, performance is a major challenge. In our KBC systems using DeepDive, we may need to perform inference and learning on a large number of highly correlated random variables. For example, in PaleoDeepDive, we construct factor graphs that contain more than 300 million variables, each representing a potential mention to extract as final output. Therefore, one of our technical focus areas has been to speed up probabilistic inference.[32, 33, 35, 50, 51] A second major technical focus has been the development of efficient *incremental* methods for grounding and inference, given the iterative nature of KBC application development. In Section 4, we briefly describe these techniques and provide pointers to readers who are interested in further details.

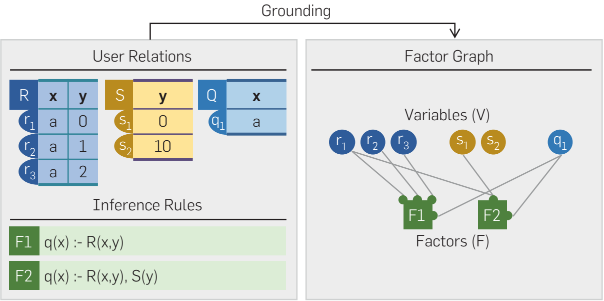
4. TECHNIQUES

A DeepDive program is a set of rules with weights specified using the language we described above. During inference, the values of all weights are assumed to be known, whereas in learning, one finds the set of weights that maximizes the probability of the evidence. The execution of a DeepDive program consists of two phases: (i) grounding and (ii) statistical inference and learning. In this section, we briefly describe the techniques we developed in each phase to make DeepDive performant and scalable.

4.1. Grounding[h]

As shown in Figure 8, DeepDive explicitly constructs a factor graph for inference and learning using a set of SQL queries. A factor graph is a triple $(V, F, \hat{w})$ in which V is a set of nodes that correspond to Boolean random variables, F is a set of hyperedges (for $f \in F$, $f \subseteq V$), and $\hat{w}: F \times \{0, 1\}^V \to \mathbb{R}$ is a weight function. In DeepDive, V and F are explicitly created using a set of SQL queries, and this process is called *grounding*.

Figure 8. Schematic illustration of grounding. Each tuple corresponds to a Boolean random variable and node in the factor graph. We create one factor for every set of groundings of an inference rule.



R	x	y
r_1	a	0
r_2	a	1
r_3	a	2

S	y
s_1	0
s_2	10

Q	x
q_1	a

F1 q(x) :- R(x,y)

F2 q(x) :- R(x,y), S(y)

[h] There is a justification for probabilistic reasoning, as Cox's theorem asserts (roughly) that if one uses numbers as degrees of belief, then one must either use probabilistic reasoning or risk contradictions in one's reasoning system; that is, a probabilistic framework is the only sound system for reasoning in this manner. We refer the reader to Jaynes et al.[21]

EXAMPLE 4.1. *Take the database instances and rules in Figure 8 as an example: each tuple in relation* R, S, *and* Q *is a random variable, and* V *contains all random variables. The inference rules* F1 *and* F2 *ground factors with the same name in the factor graph, as illustrated in Figure 8. Both* F1 *and* F2 *are implemented as SQL statements in DeepDive.*

Incremental grounding. Because DeepDive is based on SQL, we are able to take advantage of decades of work on incremental view maintenance. The input to this phase is the same as the input to the grounding phase: a set of SQL queries and the user schema. The output of this phase is how the output of grounding changes, that is, a set of modified variables ΔV and their factors ΔF. Since V and F are simply views over the database, any view maintenance techniques can be applied to incremental grounding. DeepDive uses the DRED algorithm,[17] which handles both additions and deletions. Recall that in DRED, for each relation R_i in the user's schema, we create a *delta relation*, R_i^δ, with the same schema as R_i and an additional column count. For each tuple t, t.count represents the number of derivations of t in R_i. On an update, DeepDive updates delta relations in two steps. First, for tuples in R_i^δ, DeepDive directly updates the corresponding counts. Second, an SQL query called a *delta rule* is executed, which processes these counts to generate modified variables ΔV and factors ΔF. We found that the overhead of DRED is modest and the gains may be substantial, so DeepDive always runs DRED—except on initial load.

4.2. Statistical inference and learning[i]

The main task that DeepDive conducts on factor graphs is statistical inference, that is, determining for a given node what the marginal probability is that this node takes the value 1, that it is a correct output tuple that should be included in the final knowledge base. In general, computing these marginal probabilities is #P-hard.[45] Like many other systems, DeepDive uses Gibbs sampling[40] to estimate the marginal probability of every tuple in the database.

Efficiency and scalability. There are two components to scaling statistical algorithms: *statistical efficiency*, roughly how many steps an algorithm takes to converge, and *hardware efficiency*, how efficient each of those steps is. We introduced this terminology and studied this extensively in a recent paper.[51]

DimmWitted, the statistical inference and learning engine in DeepDive,[51] is built upon our research on how to design a high-performance statistical inference and learning engine on a single machine.[27, 32, 50, 51] DimmWitted models Gibbs sampling as a "column-to-row access" operation: each row corresponds to one factor, each column to one variable, and the nonzero elements in the data matrix correspond to edges in the factor graph. To process one variable, DimmWitted fetches one column of the matrix to get the set of factors, and other columns to get the set of variables that connect to the same factor. On standard benchmarks, DimmWitted was 3.7× faster than GraphLab's implementation without any application-specific optimization. Compared with traditional work, the main novelty of DimmWitted is that it considers *both* hardware efficiency and statistical efficiency for executing an inference and learning task.

- **Hardware efficiency.** DeepDive takes into consideration the architecture of modern nonuniform memory access (NUMA) machines. A NUMA machine usually contains multiple nodes (sockets), where each socket contains multiple CPU cores. To achieve higher hardware efficiency, one wants to decrease the communication across different NUMA nodes.
- **Statistical efficiency.** Pushing hardware efficiency to the extreme might decrease statistical efficiency, because the lack of communication between nodes might decrease the rate of convergence of a statistical inference and learning algorithm. DeepDive takes advantage of the theoretical results of model averaging[53] and our own results about lock-free execution.[27, 32]

On the whole corpus of *Paleobiology*, the factor graph contains more than 0.2 billion random variables and 0.3 billion factors. On this factor graph, DeepDive is able to run Gibbs sampling on a machine with four sockets (10 cores per socket), and we find that we can generate 1000 samples for all 0.2 billion random variables in 28 min. This is more than 4× faster than a non-NUMA-aware implementation.

Incremental inference. Due to our choice of incremental grounding, the input to DeepDive's inference phase is a factor graph along with a set of changed variables and factors. The goal is to compute the output probabilities computed by the system. Our approach is to frame the incremental maintenance problem as approximate inference. Previous work in the database community has looked at how machine learning data products change in response to both new labels[24] and new data.[7, 8] In KBC, both the program and data change on each iteration. Our proposed approach can cope with both types of change simultaneously.

The technical question is which approximate inference algorithms to use in KBC applications. We choose to study two popular classes of approximate inference techniques: *sampling-based materialization* (inspired by sampling-based probabilistic databases such as MCDB[20]) and *variational-based materialization* (inspired by techniques for approximating graphical models[44]). Applying these techniques to incremental maintenance for KBC is novel, and it is not theoretically clear how the techniques compare. Thus, we conducted an experimental evaluation of these two approaches on a diverse set of DeepDive programs. We found these two approaches to be sensitive to changes along three largely orthogonal axes: the size of the factor graph, the sparsity of correlations, and the anticipated number of future changes. The performance varies by up to two orders of magnitude in different points of the space. Our study of the tradeoff space highlights that neither materialization strategy dominates the other. To automatically choose the materialization strategy, we developed a simple rule-based optimizer.[42]

[i] For example, for the grounding procedure illustrated in Figure 8, the delta rule for F1 is q(x): −R(x, y).

5. RELATED WORK

KBC has been an area of intense studies over the last decade.[2, 3, 6, 14, 23, 25, 31, 37, 41, 43, 48, 52] Within this space, there are a number of approaches.

5.1. Rule-based systems

The earliest KBC systems used pattern matching to extract relationships from text. The most well-known example is the "Hearst Pattern" proposed by Hearst[18] in 1992. In her seminal work, Hearst observed that a large number of hyponyms can be discovered by simple patterns, for example, "X such as Y." Hearst's technique has formed the basis of many further techniques that attempt to extract high-quality patterns from text. Rule-based (pattern matching-based) KBC systems, such as IBM's SystemT,[25, 26] have been built to aid developers in constructing high-quality patterns. These systems provide the user with a (declarative) interface to specify a set of rules and patterns to derive relationships. These systems have achieved state-of-the-art quality on tasks, such as parsing.[26]

5.2. Statistical approaches

One limitation of rule-based systems is that the developer needs to ensure that all rules provided to the system are high-precision rules. For the last decade, probabilistic (or machine learning) approaches have been proposed to allow the system to select from a range of a priori features automatically. In these approaches, the extracted tuple is associated with a marginal probability that it is true. DeepDive, Google's knowledge graph, and IBM's Watson are built on this approach. Within this space, there are three styles of systems based on classification,[2, 3, 6, 14, 48] maximum a posteriori,[23, 31, 43] and probabilistic graphical models.[11, 37, 52] Our work on DeepDive is based on graphical models.

6. CURRENT DIRECTIONS

6.1. Data programming

In a standard DeepDive KBC application (e.g., as in Section 3.2), the weights of the factor graph that models the extraction task are learned using either hand-labeled training data or distant supervision. However, in many applications, assembling hand-labeled training data is prohibitively expensive (e.g., when domain expertise is required), and distant supervision can be insufficient or time consuming to implement perfectly. For example, users may come up with many potential distant supervision rules that overlap, conflict, and are of varying unknown quality, and deciding which rules to include and how to resolve their overlaps could take many development cycles. In a new approach called *data programming*,[38] we allow users to specify arbitrary *labeling functions*, which subsume distant supervision rules and allow users to programatically generate training data with increased flexibility. We then learn the relative accuracies of these labeling functions and denoise their labels using automated techniques, resulting in improved performance on the KBC applications outlined.

6.2. Lightweight extraction

In some cases, users may have simple extraction tasks which need to be implemented rapidly, or may wish to first iterate on a simpler initial version of a more complex extraction task. For example, a user might have a complex extraction task involving multiple entity and relation types, connected by a variety of inference rules, over a large web-scale dataset; but they may want to start by iterating on just a single relationship over a subset of the data. For these cases, we are developing a lightweight, Jupyter notebook-based extraction system called Snorkel, intended for quick iterative development of simple extraction models using data programming.[13] We envision Snorkel as a companion and complement to DeepDive.[j]

6.3. Asynchronous inference

One method for speeding up the inference and learning stages of DeepDive is to execute them asynchronously. In recent work, we observed that asynchrony can introduce bias in Gibbs sampling, and outline some sufficient conditions under which the bias is negligible.[10] Further theoretical and applied work in this direction will allow for faster execution of complex DeepDive models asynchronously.

Acknowledgments

We gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) XDATA program under no. FA8750-12-2-0335 and DEFT program under no. FA8750-13-2-0039, DARPA's MEMEX program and SIMPLEX program, the National Science Foundation (NSF) CAREER Award under no. IIS-1353606, the Office of Naval Research (ONR) under awards nos. N000141210041 and N000141310129, the National Institutes of Health Grant U54EB020405 awarded by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative, the Sloan Research Fellowship, the Moore Foundation, American Family Insurance, Google, and Toshiba. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, AFRL, NSF, ONR, NIH, or the US government.

[j] snorkel.stanford.edu.

References

1. Angeli, G. et al. Stanford's 2014 slot filling systems. *TAC KBP* (2014).
2. Banko, M. et al. Open information extraction from the Web. In *IJCAI* (2007).
3. Betteridge, J., Carlson, A., Hong, S.A., Hruschka, E.R., Jr, Law, E.L., Mitchell, T.M., Wang, S.H. Toward never ending language learning. In *AAAI Spring Symposium* (2009).
4. Brin, S. Extracting patterns and relations from the world wide web. In *WebDB* (1999).
5. Brown, E. et al. Tools and methods for building Watson. *IBM Research Report* (2013).
6. Carlson, A. et al. Toward an architecture for never-ending language learning. In *AAAI* (2010).
7. Chen, F., Doan, A., Yang, J., Ramakrishnan, R. Efficient information extraction over evolving text data. In *ICDE* (2008).
8. Chen, F. et al. Optimizing statistical information extraction programs over evolving text. In *ICDE* (2012).
9. Chen, Y., Wang, D.Z. Knowledge expansion over probabilistic knowledge bases. In *SIGMOD* (2014).
10. De Sa, C., Olukotun, K., Ré, C. Ensuring rapid mixing and low bias for asynchronous gibbs sampling. *arXiv preprint arXiv:1602.07415* (2016).
11. Domingos, P., Lowd, D. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool, 2009.
12. Dong, X.L. et al. From data fusion to knowledge fusion. In *VLDB* (2014).
13. Ehrenberg, H.R., Shin, J., Ratner, A.J., Fries, J.A., Ré, C. Data programming with DDLite: Putting humans in a different part of the loop. In *HILDA'16 SIGMOD* (2016), 13.
14. Etzioni, O. et al. Web-scale information extraction in KnowItAll: Preliminary results. In *WWW* (2004).
15. Ferrucci, D. et al. Building Watson: An overview of the DeepQA project. *AI Magazine* (2010).

16. Govindaraju, V. et al. Understanding tables in context using standard NLP toolkits. In *ACL* (2013).
17. Gupta, A., Mumick, I.S., Subrahmanian, V.S. Maintaining views incrementally. *SIGMOD Rec.* (1993).
18. Hearst, M.A. Automatic acquisition of hyponyms from large text corpora. In *COLING* (1992).
19. Hoffmann, R. et al. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL* (2011).
20. Jampani, R. et al. MCDB: A Monte Carlo approach to managing uncertain data. In *SIGMOD* (2008).
21. Jaynes, E.T. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.
22. Jiang, S. et al. Learning to refine an automatically extracted knowledge base using Markov logic. In *ICDM* (2012).
23. Kasneci, G. et al. The YAGO-NAGA approach to knowledge discovery. *SIGMOD Rec.* (2009).
24. Koc, M.L., Ré, C. Incrementally maintaining classification using an RDBMS. *PVLDB* (2011).
25. Krishnamurthy, R. et al. SystemT: A system for declarative information extraction. *SIGMOD Rec.* (2009).
26. Li, Y., Reiss, F.R., Chiticariu, L. SystemT: A declarative information extraction system. In *HLT* (2011).
27. Liu, J. and et al. An asynchronous parallel stochastic coordinate descent algorithm. *ICML* (2014).
28. Madhavan, J. et al. Web-scale data integration: You can only afford to pay as you go. In *CIDR* (2007).
29. Mallory, E.K. et al. Large-scale extraction of gene interactions from full text literature using deepdive. *Bioinformatics* (2015).
30. Mintz, M. et al. Distant supervision for relation extraction without labeled data. In *ACL* (2009).
31. Nakashole, N. et al. Scalable knowledge harvesting with high precision and high recall. In *WSDM* (2011).
32. Niu, F. et al. Hogwild! A lock-free approach to parallelizing stochastic gradient descent. In *NIPS* (2011).
33. Niu, F. et al. Tuffy: Scaling up statistical inference in Markov logic networks using an RDBMS. *PVLDB* (2011).
34. Niu, F. et al. Elementary: Large-scale knowledge-base construction via machine learning and statistical inference. *Int. J. Semantic Web Inf. Syst.* (2012).
35. Niu, F. et al. Scaling inference for Markov logic via dual decomposition. In *ICDM* (2012).
36. Peters, S.E. et al. A machine reading system for assembling synthetic Paleontological databases. *PloS One* (2014).
37. Poon, H., Domingos, P.. Joint inference in information extraction. In *AAAI* (2007).
38. Ratner, A., De Sa, C., Wu, S., Selsam, D., Ré, C. Data programming: Creating large training sets, quickly. *arXiv preprint arXiv:1605.07723* (2016).
39. Ré, C. et al. Feature engineering for knowledge base construction. *IEEE Data Eng. Bull.* (2014).
40. Robert, C.P., Casella, G. *Monte Carlo Statistical Methods*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
41. Shen, W. et al. Declarative information extraction using datalog with embedded extraction predicates. In *VLDB* (2007).
42. Shin, J. et al. Incremental knowledge base construction using deepdive. *PVLDB* (2015).
43. Suchanek, F.M. et al. SOFIE: A self-organizing framework for information extraction. In *WWW* (2009).
44. Wainwright, M., Jordan, M. Log-determinant relaxation for approximate inference in discrete Markov random fields. *Trans. Sig. Proc.* (2006).
45. Wainwright, M.J., Jordan, M.I. Graphical models, exponential families, and variational inference. *FTML* (2008).
46. Weikum, G., Theobald, M. From information to knowledge: Harvesting entities and relationships from web sources. In *PODS* (2010).
47. Wick, M. et al. Scalable probabilistic databases with factor graphs and MCMC. *PVLDB* (2010).
48. Yates, A. et al. TextRunner: Open information extraction on the Web. In *NAACL* (2007).
49. Zhang, C. et al. GeoDeepDive: Statistical inference using familiar data-processing languages. In *SIGMOD* (2013).
50. Zhang, C., Ré, C. Towards high-throughput Gibbs sampling at scale: A study across storage managers. In *SIGMOD* (2013).
51. Zhang, C., Ré, C.. DimmWitted: A study of main-memory statistical analytics. *PVLDB* (2014).
52. Zhu, J. et al. StatSnowball: A statistical approach to extracting entity relationships. In *WWW* (2009).
53. Zinkevich, M. et al. Parallelized stochastic gradient descent. In *NIPS* (2010), 2595–2603.

Michael Cafarella and Jaeho Shin ({michael.cafarella, jaeho.shin}@lattice.io), Lattice Data, Inc., Palo Alto, CA.

Christopher De Sa, Alex Ratner, Christopher Ré, Feiran Wang, and Sen Wu ({chrismre, cdesa, ajratner, feiran, senwu}@cs.stanford.edu), Computer Science Department, Stanford University, Stanford, CA.

Ce Zhang (ce.zhang@inf.ethz.ch), ETH Zurich, Zurich, Switzerland.

[CONTINUED FROM P. 104] and interplanetary space with autonomous rovers and spacecraft. Robotic assistants and artificial intelligence are objects of intense public and academic interest and business investment. It is indeed surprising that they would not be widespread by the 24th century. Robots of various kinds are common in "Star Wars" "… long, long ago, in a galaxy far away …" Maybe it was that alternate futuristic world that tempted the creators of "Star Trek: Voyager" to finally accept a cybernetic holographic character, "the Doctor," into the crew of the starship *Voyager*.

Replicated androids like Data would make exponential cascades of robots building robots building robots … providing a workforce to render the Federation a paradise of leisure. The robots could "terraform" desolate planets and further expand the Federation. They might even build spinning space stations for artificial gravity or an inflatable planet.

Another technology being developed today is direct neural connections with electronic devices. Body hackers have surgically attached toy devices to their nervous systems, and brain-machine interfaces enjoy a flourishing research environment. Compared with direct wireless control of starship systems through thoughts alone, as relayed by, say, Bluetooth, the *Enterprise* control panels might seem insufficiently futuristic. It is but one more step to augmenting human memory and, perhaps, intellectual capability.

Some of the thousands of exoplanets that have been discovered by earthly astronomers in recent years may be ocean worlds. The Federation in the far future might thus expect to encounter floating or undersea cities in their meetings with aliens. We might then ponder the plot potential of combining another 1960s TV show, "Voyage to the Bottom of the Sea," with "Star Trek" on such a planet. The writers of the forthcoming series "Star Trek: Discovery" should keep this in mind for the sake of realism, as well as for the promise of future TV spin-offs and residuals.

Space elevators have inspired many technology lovers since Konstantin Tsiolkovsky conceived and proposed them in 1895. They offer exciting possibilities for expanding access to space at low marginal cost, along with low marginal impact on air quality from rocket exhaust. One episode of "Star Trek: Voyager" concerned a space elevator,[2] but the existence of a transporter beam apparently made the technology unnecessary. Aliens lacking a transporter beam would be interesting nonetheless. But a space elevator would have been a good alternative given the hazardous possibilities of transporter failure. You could never get me to use such a glorified Xerox machine; if the transporter malfunctions, as in "Star Trek: The Motion Picture,"[7] one could materialize as a nightmare of disorganized body parts, and worse.

The stress and awakening emotional conflict destroyed her robot mind.

Compared with medical practice in the 1960s, medical technology advanced dramatically in the "Star Trek" universe, but no scriptwriter considered the conquest of ageing and death, or immortality. Such a narrow view of the limitless "Star Trek" universe is a pity, because one would need immortality to have time to read and view all the interesting "Star Trek" and other science fiction media our own civilization is creating. My DVR is figuratively bulging with episodes of "Dark Matter" I have not had time for, and the three-part Syfy channel adaptation of Arthur C. Clarke's novel *Childhood's End*, among other titles, awaits. Science fiction productions I want to see are proliferating like Tribbles. I need the ability to absorb scenes at accelerated speed, the way Lt. Commander Gary Mitchell did when mutating into an advanced being.[10]

Only rarely did Starfleet crews pursue contact with advanced non-human beings. For example, Captain Kirk's meeting with the advanced but creepy Balok, the trippy childlike alien with adult voice in a gigantic starship led to no perceptible gains for the Federation.[12] Moreover, the Federation managed to incorporate no new classes of technology from the "new civilizations" it encountered, putting in doubt the value of the "seeking out" in the "Star Trek" slogan "… seeking out new life and new civilizations …" In "Star Trek: The Next Generation," Lieutenant Barkley undergoes a mind meld with superior aliens who were curious about humans and apparently friendly, yet had no effect on the Federation.[9] These genius beings were never seen again. Although they brought the *Enterprise* 8,000 parsecs across the Galaxy and then sent it back, the Federation never benefited from its super-warp drive; no scientific knowledge or even sources of nutrition became available. Maybe if the beings were warlike and dangerous to humans, the screenwriters would have found them more compelling and followed up. Another genius civilization, the Q beings, had, however, transcended technology and viewed the Federation with contempt. The story of how the Q achieved their transcendence would have been fascinating. What might we gain if we really did contact advanced civilizations, unconstrained by the boundaries of an episodic weekly TV show? Imagine the possibilities …

References

1. Bixby, J. *Requiem for Methuselah.* Star Trek episode 74; https://en.wikipedia.org/wiki/Requiem_for_Methuselah
2. Braga, B. (screenplay) and Diggs, J. (story). *Rise.* Star Trek: Voyager episode 61; https://en.wikipedia.org/wiki/Rise_(Star_Trek:_Voyager)
3. Fontana, D.C. (teleplay) and Wolfe, L.N. (story). *The Ultimate Computer.* Star Trek episode 53; https://en.wikipedia.org/wiki/The_Ultimate_Computer
4. Gerrold, D. *The Trouble with Tribbles.* Star Trek episode 44; https://en.wikipedia.org/wiki/The_Trouble_with_Tribbles
5. Kandel, S. *I, Mudd.* Star Trek episode 37; https://en.wikipedia.org/wiki/I,_Mudd
6. Leinster, M. A logic named Joe. *Astounding Science Fiction* (Mar. 1946).
7. Livingston, H. (screenplay) and Foster, A.D. (story). *Star Trek: The Motion Picture*; https://en.wikipedia.org/wiki/Star_Trek:_The_Motion_Picture
8. Lucas, J.M. *The Changeling.* Star Trek episode 32; https://en.wikipedia.org/wiki/The_Changeling_(Star_Trek:_The_Original_Series)
9. Menosky, J. *The Nth Degree.* Star Trek The Next Generation episode 93; https://en.wikipedia.org/wiki/The_Nth_Degree_(Star_Trek:_The_Next_Generation)
10. Peeples, S.A. *Where No Man Has Gone Before.* Star Trek episode 3; https://en.wikipedia.org/wiki/Where_No_Man_Has_Gone_Before
11. Schneider, P. *Balance of Terror.* Star Trek episode 14; https://en.wikipedia.org/wiki/Balance_of_Terror
12. Sohl, J. *The Corbomite Maneuver.* Star Trek episode 10; https://en.wikipedia.org/wiki/The_Corbomite_Maneuver
13. Sowards, J.B. (screenplay) and Bennett, H. and Sowards, J.B. (story). *Star Trek II: The Wrath of Khan*; https://en.wikipedia.org/wiki/Star_Trek_II:_The_Wrath_of_Khan
14. Sternbach, R. and Okuda, M. *Star Trek The Next Generation Technical Manual.* Pocket Books, New York, 1991.

David Allen Batchelor (batchelor@alum.mit.edu) is a scientist and computer engineer for data systems at NASA Goddard Space Flight Center, Greenbelt, MD. His first science-fiction novel, *The Metalmark Contract*, was published in 2011 by Black Rose Writing, Castroville, TX.

From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/3069382

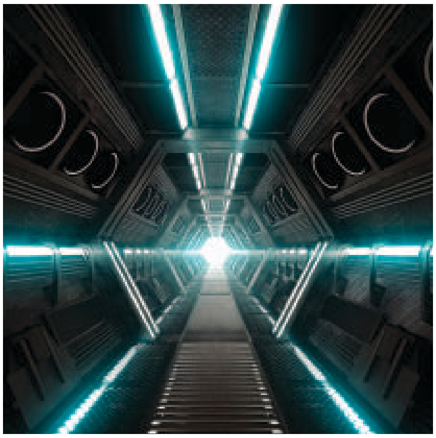
David Allen Batchelor

Future Tense
Beyond ‘Star Trek’

On a mission to boldly go where no man has gone before, the series and movies somehow missed some promising technologies …

THE 50th ANNIVERSARY in 2016 of the iconic franchise saw multiple checklists of the speculative technologies that have become real due in part to the inspiring vision of “Star Trek,” and many fans continue to cheer for even more treknology. This is an amazing record for a low-budget 1960s TV show (approximately $190,700 per episode) that first struggled for ratings but then spawned three subsequent “Star Trek” TV series and 13 movies.[14] “Star Trek” has inspired technological innovation from smartphones to quantum physics, and the enduring popularity of the original show in syndication continues to make it a launchpad for future ideas and advances “Star Trek” creators never imagined. Here, I explore some of the technologies, from simple to far-out, that might yet find a place in the “Star Trek” universe if the forthcoming series gets the budget and ratings it deserves.

Many technologies would have improved the “Star Trek” universe in terms of realism and physical common sense. Some, like seatbelts, are simple and primitive, and would have kept numerous crewmembers assigned to the bridge from being shaken up when the starship *Enterprise* took a hit from, say, a Romulan plasma torpedo.[11] As another example, when furry, prolific Tribbles experienced a population explosion aboard the *Enterprise*,[4] they could have become a nourishing resource for a remote Federation colony, not sent to some horrific fate aboard a Klingon battle cruiser. What if, instead of consuming Tribbles, Federation scientists had genetically enhanced them for intellect? A talking Tribble or other bioengineered alien could offer companionship and amusing views in a future universe of starships some human viewers found sterile.



Early “Star Trek” scriptwriters did not anticipate a network of computers, even though, in 1946, science fiction writer Murray Leinster predicted a worldwide Internet-like network in his story “A Logic Named Joe.”[6] Social networks are not a feature of computer use in the “Star Trek” universe. The writers stuck with isolated mainframes like the ship’s computer, even though such monolithic machines went awry, as with the M5 multitronic unit,[3] or were hacked by more computationally advanced aliens, invaded and pwned. A shipboard network of special-purpose processors might be less vulnerable.

Telepresence robots today let us explore space and deep-ocean environments, perform remote surgery, visit the insides of malfunctioning nuclear power plants, and disarm bombs. Such a device might have spared the life of Mr. Spock when he sacrificed himself to save the *Enterprise* by repairing its radioactive warp engine in the heart-rending death scene in the movie *Star Trek II: The Wrath of Khan*.[13]

The great project of artificial intelligence, begun in earnest in the 20th century, foundered in the world of the original “Star Trek” series. If artificial humanoids were encountered, they were threats or fatally flawed. In one episode, Nomad, a robotic space probe, returned from its mission with newfound destructive intent.[8] In another, the robot colony that captured Harry Mudd, led by its chief, Norman,[5] decided to seduce all humanity with offers of service and had to be subdued with illogical assertions and paradoxes. Afterward, the tamed robots were left to themselves, as if for them to serve any useful role would have been a disturbance in the established order of the Federation. Another episode featured a humanoid robot woman called Rayna who was deceived by her maker into believing she was human,[1] but Captain Kirk ruined that project by attracting her to himself and forcing her to confront her beloved creator. The stress and awakening emotional conflict destroyed her robot mind. In the universe of 23rd-century “Star Trek,” that particular AI project seemed ill conceived. Maybe the scriptwriters feared the robots would rebel and go into business for themselves, as with the Nomad probe.

In the 24th-century environment of the “Star Trek: The Next Generation” TV series, robots would be even more scarce than before, except for *Enterprise* crew member Data, who seemed to be an isolated experiment. NASA today explores Mars

[CONTINUED ON P. 103]

IMAGE FROM SHUTTERSTOCK.COM

DOI:10.1145/3072528

Bo Brinkman, Catherine Flick, Don Gotterbarn,[a] Keith Miller, Kate Vazansky, Marty J. Wolf

Listening to Professional Voices: Draft 2 of the ACM Code of Ethics and Professional Conduct

FOR THE FIRST time since 1992, the ACM Code of Ethics and Professional Conduct (the Code) is being updated. The Code Update Task Force in conjunction with the Committee on Professional Ethics is seeking advice from ACM members on the update. We indicated many of the motivations for changing the Code when we shared Draft 1 of Code 2018 with the ACM membership in the December 2016 issue of CACM[b] and with others through email and the COPE website (ethics.acm.org). Since December, we have been collecting feedback and are vetting proposed changes.

We have seen a broad range of concerns about responsible computing including bullying in social media, cyber security, and autonomous machines making ethically significant decisions. The Task Force appreciates the many serious and thoughtful comments it has received. In response, the Task Force has proposed changes that are reflected in Draft 2 of the Code. There are a number of substantial changes that require some explanation. In this article, we discuss these, and we explain why we did not include other requested changes in Draft 2. We look forward to receiving your comments on these suggested changes and your requests for additional changes as we work on Draft 3 of the Code. We have provided opportunities for your comments and an open discussion of Draft 2 at the ACM Code 2018 Discussion website [http://code2018.acm.org/discuss]. Comments can also be contributed at the COPE website https://ethics.acm.org, and by direct emails to chair@ethics.acm.org.

The Nature of an Ethics Code

ACM members are part of the computing profession and the ACM's Code of Ethics and Professional Conduct should reflect the conscience of the computing profession. When the Code adequately reflects the ethics of the profession, it also clarifies what that profession should strive to be. A code provides positive direction for its members.

The current update of the ACM Code begins positively; "Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing." As computing professionals, we are asked to promote good while working within ethical constraints including: be honest, don't cause harm, and avoid conflicts of interest. As the areas in which computing can make a positive impact have increased so has the range of our moral responsibility.

In Draft 1, the Task Force's suggested modifications reflected the need for members to better understand how computing technologies and artifacts impact the social infrastructure and how they ought to promote the common good. Professionalism in computing requires us to improve our abilities to anticipate broader impacts, both positive and negative, and to accept responsibility for those impacts.

This understanding of a code helps address concerns expressed by many commenters who noted a lack of clarity about to whom the ACM's Code applies. There were places where the Code seemed to apply to computing professionals more generally and other places where it seemed to apply only to ACM members. There were even a few places where the Code seemed to apply only to ACM members who were also computing professionals.

These concerns are addressed in Draft 2 in three ways. First, the Preamble now identifies what is meant by "computing professional." We intend for this term to be interpreted broadly, including students, software engineers, software architects, managers, leaders, and computer science teachers and scholars. Given the ubiquity of computing and the aspirational nature of the Code, we therefore aim to include those who may consider themselves professionals in the area of computing from non-standard backgrounds as well as those more traditionally considered computing professionals.

A second change intended to reflect that the Code provides aspirational guidance to a broad community involved replacing the categorical language of "moral imperatives" with the less prescriptive "ethical principles." Each of the principles in the Code is to be used to help us understand our ethical responsibility and to guide our decision making in varying and complex situations, rather than provide a rigid set of rules to follow unthinkingly. These principles are to be considered in our deliberations as we set professional goals for ourselves and carry out our daily activities. Section 1, especially, sets forth principles that need to be given special weight in those deliberations.

A third change was to clarify that every principle applies to computing professionals, regardless of their affiliation with the ACM, with the exception of the guidance given in Section 4. In principle 4.1, ACM members take on the additional responsibility of encouraging and supporting adherence to the ACM Code by all computing professionals. In the guidance for principle 4.2, we have retained the language whereby ACM members who violate the Code may have their membership terminated.

Requested Changes Made

One of the primary reasons for updating the Code is the increased influ-

[a] Corresponding author and chair of Code 2018 project chair@ethics.acm.org

[b] http://cacm.acm.org/magazines/2016/12/210366-the-acm-code-of-ethics/fulltext

ence of computing since 1992. Principle 1.1 has been modified to make this change (that almost all people are now impacted by computing) explicit by adding to the principle "acknowledging that all people are stakeholders in computing and its artifacts." The phrase "computing and its artifacts" is meant to remind practitioners that it is not just the code that they write that matters, but also those things that emerge from that code. In particular, the Task Force is addressing growing concerns about algorithms that emerge from machine learning rather than directly from algorithm designers. Consistent with the importance of computing and the ways it can contribute to society, we added an encouragement to perform pro bono or volunteer work. Like other professions, computing is a service to society. Following John Rawls' difference principle,[c] we emphasized computing professionals' responsibility toward the least powerful: "When the interest of multiple groups conflict, the needs of the least advantaged should be given increased attention and priority."

The revisions to principle 1.2 continue the clarification of a computing professional's responsibility to a broad range of stakeholders, and of the responsibility not to harm them. Sometimes causing harm is not unethical; examples often cited include self-defense and a just war. We have modified this principle to reflect these exceptions. Emergent technologies such as data remixing or policy-making software can also cause harm. To address this concern we have added, "Those involved with pervasive or infrastructure systems should also consider Principle 3.7" which advocates deeper analysis of emergent systems such as machine learning.

In the 1992 Code, principle 1.4 read "Be fair and take action not to discriminate." There was some concern that this might be misinterpreted due to the fact that "discrimination" does not necessarily imply unfairness, and in Draft 1 it was changed to "Be fair and take action not to discriminate *unfairly*." This has been roundly criticized as being even worse and may appear to some as a loophole for those who are seeking to justify discrimination that is unfair. Hence, in Draft 2, we have reverted back to the 1992 language.

Professionalism in computing requires us to improve our abilities to anticipate broader impacts, both positive and negative, and to accept responsibility for those impacts.

A frequent request was to explicitly address harassment, and especially sexual harassment, in the Code. The line "Sexual harassment is a form of discrimination that limits fair access to the spaces where the harassment takes place" has been added to the guidance of principle 1.4. The Task Force is attempting to correct a common misunderstanding about sexual harassment, the (false) belief that it does not have any consequences beyond just offending the harassed party. Instead, we emphasize that harassment is also a form of unfair discrimination because it makes the workplace or place of study unfairly inhospitable to certain individuals based on their identity. Sexual harassment is, in itself, an offense against principle 1.4 and other principles of the Code.

Principle 1.4 also speaks against bullying, a form of harassment based on a power differential rather than on sexual difference (although sexual harassment may also include power differentials). For example, it speaks against academic bullying which may occur when a more established scholar, or a person who has power because of their position (for example, an editor or program committee member), misuses that power to make unreasonable demands or to harm early career scholars, including graduate students. Bullying is also a form of unfair discrimination, as it does not recognize the inherent worth of every person and group.

In the 1992 Code, principles 1.5 and 1.6 were about honoring physical and intellectual property (IP) rights (copyright, patents, and crediting others' work). Draft 1 merged these to create a single statement about intellectual property rights. Understandably, the world of IP has changed significantly since 1992, and these days the definitions of "intellectual property" are complex and controversial, particularly in the computing world. Thus we received some significant criticism on the rewrite of this section, from multiple sides of intellec-

[c] Rawls, J. (2001) *Justice as Fairness: A Restatement*, E. Kelly (ed.), Cambridge, MA: Harvard University Press.

tual property arguments. The Task Force, after extensive discussion both internally and externally,[d] simplified the focus of this principle to a basic concept that the Code should protect the time, effort, and often considerable risk taken by people who come up with new ideas, innovations, and creative works; computing professionals should honor those investments. These creators usually have made decisions about how to protect their work; choices include open source or creative commons licensing, copyright, patents, other traditional legal avenues, or wanting no protection at all. This change also takes into account norms for specific endeavors, for example, the expectation that academic work will be cited if used in other research, teaching, or innovation. Since created works add significant value to society, the Code specifies that the creators' wishes for their works should be respected.

In moving away from explicitly listing in the Code the specific methods to be respected with respect to intellectual property works, we allow for a continuing dialogue on what the legal methods ought to be and focus on what computing professionals should do once a method is decided upon by the creator. We hope that computing professionals will be encouraged to investigate more open methods of sharing their works, with the full knowledge that the Code requires other professionals to respect their decisions about their works.

In addition to these requested changes, the Task Force made a number of smaller changes. For example, the guidance to principle 2.6 was shortened in order to add clarity. To further emphasize the importance of using the public good as the paramount decision-making principle, we moved principle 3.4 to principle 3.1, which resulted in the renumbering of principles 3.1, 3.2, and 3.3. We made further clarifying changes in principles 3.2, 3.3, and 3.4 to better reflect that leaders and groups in contemporary software development process are often more flexible and transient.

Requested Changes Not Made

There were numerous requests for more specificity within the Code. Many commenters were looking for clear and specific definitions of terms like "harm" and "public good." Presumably, with more detailed definitions of these terms, there would be more clarity about applying the Code to specific situations. That is, the Code would become much more like an algorithm that would generate a clear indication of required action in specific situations. We decided against this request primarily for two reasons. The first is to reflect that society and social values are fluid. Our second reason stems from the fact that one of the responsibilities established in principle 2.2 is for the computing professional to maintain "skill in reflective analysis for recognizing and navigating ethical challenges." A computing professional who is maintaining such skill will quite naturally be in a position to understand these more fluid terms. Indeed, part of professional practice might include regular reflection on the nature of these terms.

Additionally, there were requests to incorporate into the Code explicit principles and guidance relating to specific forms of computing technology such as cyber security and artificial intelligence. While it is clear that these are areas of concern, they are beyond the scope of a code of ethics that is intended for the more broad definition of "computing professional" that we employ here. The particular ethical behaviors surrounding specific computing technologies are derivable from the general principles of the Code. For example, it follows from principle 2.5 that those working in AI should do a proactive analysis of the potential future impacts of self-mutating code. Nonetheless, COPE is planning on developing supporting materials that will illustrate how these broad principles apply to specific technologies. In our experience, changing supporting materials is far easier than changing the Code, so this strategy should help the ACM to be more agile in reacting to the ethical implications of new applications of technology.

Finally, there were also requests for including a compliance policy in the Code. The Task Force has chosen to approach compliance in a different way. The Code is something that can be used by all computing professionals regardless of their affiliation with the ACM, but compliance issues are limited to ACM members and ACM events. Therefore, aside from the broad principles in Section 4, compliance procedures will be in the ACM bylaws, not in the Code itself. COPE is cooperating with the ACM Council to develop a new compliance policy that better supports enforcement of the Code. We plan to include in those policies appropriate due process procedures, and multiple levels of sanctions to better reflect that some violations of the Code are more serious than others.

We invite further suggestions on issues that COPE might consider for future revisions. They can be submitted at the ACM Code 2018 Discussion website (http://code2018.acm.org/discuss) We look forward to receiving your comments for improving the Code.

ACM Code of Ethics and Professional Conduct: Draft 2

Draft 2 was developed by The Code 2018 Task Force. (It is based on the 2018 ACM Code of Ethics and Professional Conduct: Draft 1).[e]

Preamble

The ACM Code of Ethics and Professional Conduct ("the Code") identifies key elements of ethical conduct in computing.

The Code is designed to support all computing professionals, which is taken to mean current or aspiring computing practitioners as well as those who influence their professional development, and those who use technology in an impactful way. The Code includes principles formulated as statements of responsibility, based on the understanding that the public good is always a primary consideration. Section 1 outlines fundamental ethical considerations. Section 2 addresses additional, more

[d] The Task Force would like to thank Brian Ballsun-Stanton in particular for his feedback on an early redraft of this section.

[e] A complete track changes version of Draft 2 showing all additions and deletions to Draft 1 version is available at http://ethics.acm.org/code-2018.

specific considerations of professional responsibility. Section 3 pertains more specifically to individuals who have a leadership role, whether in the workplace or in a volunteer professional capacity. Commitment to ethical conduct is required of every ACM member and principles involving compliance with the Code are given in Section 4.

The Code as a whole is concerned with how fundamental ethical principles apply to one's conduct as a computing professional. Each principle is supplemented by guidelines, which provide explanations to assist members in understanding and applying it. These extraordinary ethical responsibilities of computing professionals are derived from broadly accepted ethical principles.

The Code is not an algorithm for solving ethical problems, rather it is intended to serve as a basis for ethical decision making in the conduct of professional work. Words and phrases in a code of ethics are subject to varying interpretations, and a particular principle may seem to conflict with other principles in specific situations. Questions related to these kinds of conflicts can best be answered by thoughtful consideration of the fundamental ethical principles, understanding the public good is the paramount consideration. The entire profession benefits when the ethical decision making process is transparent to all stakeholders. In addition, it may serve as a basis for judging the merit of a formal complaint pertaining to a violation of professional ethical standards.

1. GENERAL MORAL PRINCIPLES

A computing professional should…

1.1 Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.

This principle concerning the quality of life of all people affirms an obligation to protect fundamental human rights and to respect diversity. An essential aim of computing professionals is to minimize negative consequences of computing, including threats to health, safety, personal security, and privacy. Computing professionals should give consideration to whether the products of their efforts will be used in socially responsible ways, will meet social needs, and will be broadly accessible. They are encouraged to actively contribute to society by engaging in pro bono or volunteer work. When the interests of multiple groups conflict the needs of the least advantaged should be given increased attention and priority.

In addition to a safe social environment, human well-being requires a safe natural environment. Therefore, computing professionals should be alert to, and make others aware of, any potential harm to the local or global environment.

1.2 Avoid harm.

In this document, "harm" means negative consequences to any stakeholder, especially when those consequences are significant and unjust. Examples of harm include unjustified death, unjustified loss of information, and unjustified damage to property, reputation, or the environment. This list is not exhaustive.

Well-intended actions, including those that accomplish assigned duties, may unexpectedly lead to harm. In such an event, those responsible are obligated to undo or mitigate the harm as much as possible. Avoiding unintentional harm begins with careful consideration of potential impacts on all those affected by decisions.

To minimize the possibility of indirectly harming others, computing professionals should follow generally accepted best practices for system design, development, and testing. Additionally, the consequences of emergent systems and data aggregation should be carefully analyzed. Those involved with pervasive or infrastructure systems should also consider Principle 3.7.

At work, a computing professional has an additional obligation to report any signs of system risks that might result in serious personal or social harm. If one's superiors do not act to curtail or mitigate such risks, it may be necessary to "blow the whistle" to reduce potential harm. However, capricious or misguided reporting of risks can itself be harmful. Before reporting risks, the computing professional should thoroughly assess all relevant aspects of the incident as outlined in Principle 2.5.

1.3 Be honest and trustworthy.

Honesty is an essential component of trust. A computing professional should be fair and not make deliberately false or misleading claims and should provide full disclosure of all pertinent system limitations and potential problems. Fabrication of data, falsification of data, and scientific misconduct are similarly violations of the Code. One who is professionally dishonest is accountable for any resulting harm.

A computing professional should be honest about his or her own qualifications, and about any limitations in competence to complete a task. Computing professionals should be forthright about any circumstances that might lead to conflicts of interest or otherwise tend to undermine the independence of their judgment.

Membership in volunteer organizations such as ACM may at times place individuals in situations where their statements or actions could be interpreted as carrying the "weight" of a larger group of professionals. An ACM member should exercise care not to misrepresent ACM, or positions and policies of ACM or any ACM units.

1.4 Be fair and take action not to discriminate.

The values of equality, tolerance, respect for others, and equal justice govern this principle. Prejudicial discrimination on the basis of age, color, disability, ethnicity, family status, gender identity, military status, national origin, race, religion or belief, sex, sexual orientation, or any other inappropriate factor is an explicit violation of ACM policy. Sexual harassment is a form of discrimination that limits fair access to the spaces where the harassment takes place.

Inequities between different groups of people may result from the use or misuse of information and technology. Technologies should be as inclusive and accessible as possible. Failure to design for inclusiveness and accessibility may constitute unfair discrimination.

1.5 Respect the work required to produce new ideas, inventions, and other creative and computing artifacts.
The development of new ideas, inventions, and other creative and computing artifacts creates value for society, and those who expend the effort needed for this should expect to gain value from their work. Computing professionals should therefore provide appropriate credit to the creators of ideas or work. This may be in the form of respecting authorship, copyrights, patents, trade secrets, non-disclosure agreements, license agreements, or other methods of attributing credit where it is due.

Both custom and the law recognize that some exceptions to a creator's control of a work are necessary to facilitate the public good. Computing professionals should not unduly oppose reasonable uses of their intellectual works.

Efforts to help others by contributing time and energy to projects that help society illustrate a positive aspect of this principle. Such efforts include free and open source software and other work put into the public domain. Computing professionals should avoid misappropriation of a commons.

1.6 Respect privacy.
”Privacy” is a multi-faceted concept and a computing professional should become conversant in its various definitions and forms.

Technology enables the collection, monitoring, and exchange of personal information quickly, inexpensively, and often without the knowledge of the people affected. Computing professionals should use personal data only for legitimate ends and without violating the rights of individuals and groups. This requires taking precautions to ensure the accuracy of data, as well as protecting it from unauthorized access or accidental disclosure to inappropriate individuals or groups. Computing professionals should establish procedures that allow individuals to review their personal data, correct inaccuracies, and opt out of automatic data collection.

Only the minimum amount of personal information necessary should be collected in a system. The retention and disposal periods for that information should be clearly defined and enforced, and personal information gathered for a specific purpose should not be used for other purposes without consent of the individual(s). When data collections are merged, computing professionals should take special care for privacy. Individuals may be readily identifiable when several data collections are merged, even though those individuals are not identifiable in any one of those collections in isolation.

“Privacy” is a multi-facet concept and a computing professional should become conversant in its various definitions and forms.

1.7 Honor confidentiality.
Computing professionals should protect confidentiality unless required to do otherwise by a bona fide requirement of law or by another principle of the Code.

User data observed during the normal duties of system operation and maintenance should be treated with strict confidentiality, except in cases where it is evidence for the violation of law, of organizational regulations, or of the Code. In these cases, the nature or contents of that information should not be disclosed except to appropriate authorities, and the computing professional should consider thoughtfully whether such disclosures are consistent with the Code.

2. PROFESSIONAL RESPONSIBILITIES

A practicing computing professional should...

2.1 Strive to achieve the highest quality in both the process and products of professional work.
Computing professionals should insist on high quality work from themselves and from colleagues. This includes respecting the dignity of employers, colleagues, clients, users, and anyone affected either directly or indirectly by the work. High quality process includes an obligation to keep the client or employer properly informed about progress toward completing that project. Professionals should be cognizant of the serious negative consequences that may result from poor quality and should resist any inducements to neglect this responsibility.

2.2 Maintain high standards of professional competence, conduct, and ethical practice.
High quality computing depends on individuals and teams who take personal and organizational responsibility for acquiring and maintaining professional competence. Professional competence starts with technical knowledge and awareness of the social context in which the work may be deployed. Professional competence also requires skill in reflective analysis for recognizing and navigating ethical challenges. Upgrading necessary skills should be ongoing and should include independent study, conferences, seminars, and other informal or formal education. Professional organizations, including ACM, are committed to encouraging and facilitating those activities.

2.3 Know, respect, and apply existing laws pertaining to professional work.
ACM members must obey existing regional, national, and international laws unless there is a compelling ethical justification not to do so. Policies and procedures of the organizations in which one participates must also be obeyed, but compliance must be balanced with the recognition that sometimes existing laws and rules are immoral or inappropriate and, therefore, must be challenged. Violation of a law or regulation may be ethical when that law or rule has inadequate moral basis or when it conflicts with another law judged to be more important. If one decides to violate a law or rule because it is unethical, or for any other reason, one must fully accept responsibility for one's actions and for the consequences.

2.4 Accept and provide appropriate professional review.
Quality professional work in computing depends on professional reviewing and critiquing. Whenever appropriate, computing professionals should seek and utilize peer and stakeholder review. Computing professionals should also provide constructive, critical review of the work of others.

2.5 Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.
Computing professionals should strive to be perceptive, thorough, and objective when evaluating, recommending, and presenting system descriptions and alternatives. Computing professionals are in a position of special trust, and therefore have a special responsibility to provide objective, credible evaluations to employers, clients, users, and the public. Extraordinary care should be taken to identify and mitigate potential risks in self-changing systems. Systems whose future risks are unpredictable require frequent reassessment of risk as the system develops or should not be deployed. When providing evaluations the professional must also identify any relevant conflicts of interest, as stated in Principle 1.3.

As noted in the guidance for Principle 1.2 on avoiding harm, any signs of danger from systems should be reported to those who have opportunity and/or responsibility to resolve them. See the guidelines for Principle 1.2 for more details concerning harm, including the reporting of professional violations.

2.6 Accept only those responsibilities for which you have or can obtain the necessary expertise, and honor those commitments.
A computing professional has a responsibility to evaluate every potential work assignment. If the professional's evaluation reveals that the project is infeasible, or should not be attempted for other reasons, then the professional should disclose this to the employer or client, and decline to attempt the assignment in its current form.

Once it is decided that a project is feasible and advisable, the professional should make a judgment about whether the project is appropriate to the professional's expertise. If the professional does not currently have the expertise necessary to complete the project the professional should disclose this shortcoming to the employer or client. The client or employer may decide to pursue the project with the professional after time for additional training, to pursue the project with someone else who has the required expertise, or to forego the project.

The major underlying principle here is the obligation to accept personal accountability for professional work. The computing professional's ethical judgment should be the final guide in deciding whether to proceed.

2.7 Improve public understanding of computing, related technologies, and their consequences.
Computing professionals have a responsibility to share technical knowledge with the public by creating awareness and encouraging understanding of computing, including the impacts of computer systems, their limitations, their vulnerabilities, and opportunities that they present. This imperative implies an obligation to counter any false views related to computing.

2.8 Access computing and communication resources only when authorized to do so.
This principle derives from Principle 1.2 - "Avoid harm to others." No one should access or use another's computer system, software, or data without permission. One should have appropriate approval before using system resources, unless there is an overriding concern for the public good. To support this clause, a computing professional should take appropriate action to secure resources against unauthorized use. Individuals and organizations have the right to restrict access to their systems and data so long as the restrictions are consistent with other principles in the Code (such as Principle 1.4).

3. PROFESSIONAL LEADERSHIP PRINCIPLES

In this section, "leader" means any member of an organization or group who has influence, educational responsibilities, or managerial responsibilities. These principles generally apply to organizations and groups, as well as their leaders.

A computing professional acting as a leader should…

3.1 Ensure that the public good is a central concern during all professional computing work.
The needs of people—including users, other people affected directly and

indirectly, customers, and colleagues—should always be a central concern in professional computing. Tasks associated with requirements, design, development, testing, validation, deployment, maintenance, end-of-life processes, and disposal should have the public good as an explicit criterion for quality. Computing professionals should keep this focus no matter which methodologies or techniques they use in their practice.

3.2 Articulate, encourage acceptance of, and evaluate fulfillment of the social responsibilities of members of an organization or group.
Technical organizations and groups affect the public at large, and their leaders should accept responsibilities to society. Organizational procedures and attitudes oriented toward quality, transparency, and the welfare of society will reduce harm to members of the public and raise awareness of the influence of technology in our lives. Therefore, leaders should encourage full participation in meeting social responsibilities and discourage tendencies to do otherwise.

3.3 Manage personnel and resources to design and build systems that enhance the quality of working life.
Leaders are responsible for ensuring that systems enhance, not degrade, the quality of working life. When implementing a system, leaders should consider the personal and professional development, accessibility, physical safety, psychological well-being, and human dignity of all workers. Appropriate human-computer ergonomic standards should be considered in system design and in the workplace.

3.4 Establish appropriate rules for authorized uses of an organization's computing and communication resources and of the information they contain.
Leaders should clearly define appropriate and inappropriate uses of organizational computing resources. These rules should be clearly and effectively communicated to those using their computing resources. In addition, leaders should enforce those rules, and take appropriate action when they are violated.

3.5 Articulate, apply, and support policies that protect the dignity of users and others affected by computing systems and related technologies.
Dignity is the principle that all humans are due respect. This includes the general public's right to autonomy in day-to-day decisions.

Designing or implementing systems that deliberately or inadvertently violate, or tend to enable the violation of, the dignity or autonomy of individuals or groups is ethically unacceptable. Leaders should verify that systems are designed and implemented to protect dignity.

3.6 Create opportunities for members of the organization and group to learn, respect, and be accountable for the principles, limitations, and impacts of systems.
This principle complements Principle 2.7 on public understanding. Educational opportunities are essential to facilitate optimal participation of all organization or group members. Leaders should ensure that opportunities are available to computing professionals to help them improve their knowledge and skills in professionalism, in the practice of ethics, and in their technical specialties, including experiences that familiarize them with the consequences and limitations of particular types of systems. Professionals should know the dangers of oversimplified models, the improbability of anticipating every possible operating condition, the inevitability of software errors, the interactions of systems and the contexts in which they are deployed, and other issues related to the complexity of their profession.

3.7 Recognize when computer systems are becoming integrated into the infrastructure of society, and adopt an appropriate standard of care for those systems and their users.
Organizations and groups occasionally develop systems that become an important part of the infrastructure of society. Their leaders have a responsibility to be good stewards of that commons. Part of that stewardship requires that computing professionals monitor the level of integration of their systems into the infrastructure of society. As the level of adoption changes, there are likely to be changes in the ethical responsibilities of the organization. Leaders of important infrastructure services should provide due process with regard to access to these services. Continual monitoring of how society is using a product will allow the organization to remain consistent with their ethical obligations outlined in the principles of the code. Where such standards of care do not exist, there may be a duty to develop them.

4. COMPLIANCE WITH THE CODE

A computing professional should…

4.1 Uphold, promote, and respect the principles of the Code.
The future of computing depends on both technical and ethical excellence. Computing professionals should adhere to the principles expressed in the Code. Each ACM member should encourage and support adherence by all computing professionals. Computing professionals who recognize breaches of the Code should take whatever actions are within their power to resolve the ethical issues they recognize.

4.2 Treat violations of the Code as inconsistent with membership in ACM.
If an ACM member does not follow the Code, membership in ACM may be terminated.

Join the Discussion

The Committee on Professional Ethics is asking you to participate in an open discussion about this Code and suggest ways in which it might be improved: **http://code2018.acm.org/discuss**; **https://ethics.acm.org**; or by direct email to **chair@ethics.acm.org**.

Bo Brinkman (bo.brinkman@miamioh.edu) is an associate professor of computer science and software engineering at Miami University, Oxford, OH.

Catherine Flick (cflick@dmu.ac.uk) is a Senior Lecturer in Computing and Social Responsibility at De Montfort University, Leicester, UK.

Don Gotterbarn (gotterbarn@acm.org) is chair of the ACM Committee on Professional Ethics and Professor Emeritus in the Department of Computing at East Tennessee State University, Johnson City.

Keith Miller (millerkei@umsl.edu) is the Orthwein Endowed Professor for Lifelong Learning in the Sciences College of Education, University of Missouri, St. Louis.

Kate Vazansky (kate.vazansky@gmail.com) is a Technical Program Manager at Salesforce.

Marty J. Wolf (mjwolf@acm.org) is a professor of computer science at Bemidji State University, Bemidji, MN.

VRST 2017

23rd ACM Symposium on Virtual Reality Software and Technology

8–10 November 2017, Gothenburg, Sweden

Full and Short Papers:

Title/abstract:	30 June 2017
Materials upload:	07 July 2017
Notification:	28 August 2017
Camera-ready:	18 September 2017

Posters and Demos:

Materials upload:	18 August 2017
Notification:	11 September 2017
Camera-ready:	18 September 2017

The ACM Symposium on Virtual Reality Software and Technology (VRST) is an international forum for the exchange of experience and knowledge among researchers and developers concerned with augmented and virtual reality (AR/VR) software and technology.

VRST 2017 will provide an opportunity for AR/VR researchers and industry to interact, share new results, show live demonstrations of their work, and discuss emerging directions for the field. The event is sponsored by ACM SIGCHI and SIGGRAPH.

Symposium Chairs: Morten Fjeld (Chalmers University of Technology), Daniel Sjölie (University of Gothenburg), and Marco Fratarcangeli (Chalmers University of Technology).

http://vrst.acm.org/vrst2017/

acm | SIGCHI special interest group computer human interaction | ACMSIGGRAPH | CHALMERS UNIVERSITY OF TECHNOLOGY | LINDHOLMEN SCIENCE PARK | Visual Arena

ACM ISS 2017

Interactive Surfaces and Spaces
October 17-20 | Brighton, UK

Submissions

19 May 2017	Workshops & Tutorials
28 June 2017	Papers & Notes (Abstracts)
4 July 2017	Papers & Notes (Full Submissions)
19 July 2017	Posters, Demos, Arts & Videos
1 August 2017	Doctoral Symposium

http://iss2017.acm.org