Making sense of AI

Guest

Why machine programming should be the next technology you invest in

Justin Gottschlich, Intel Labs
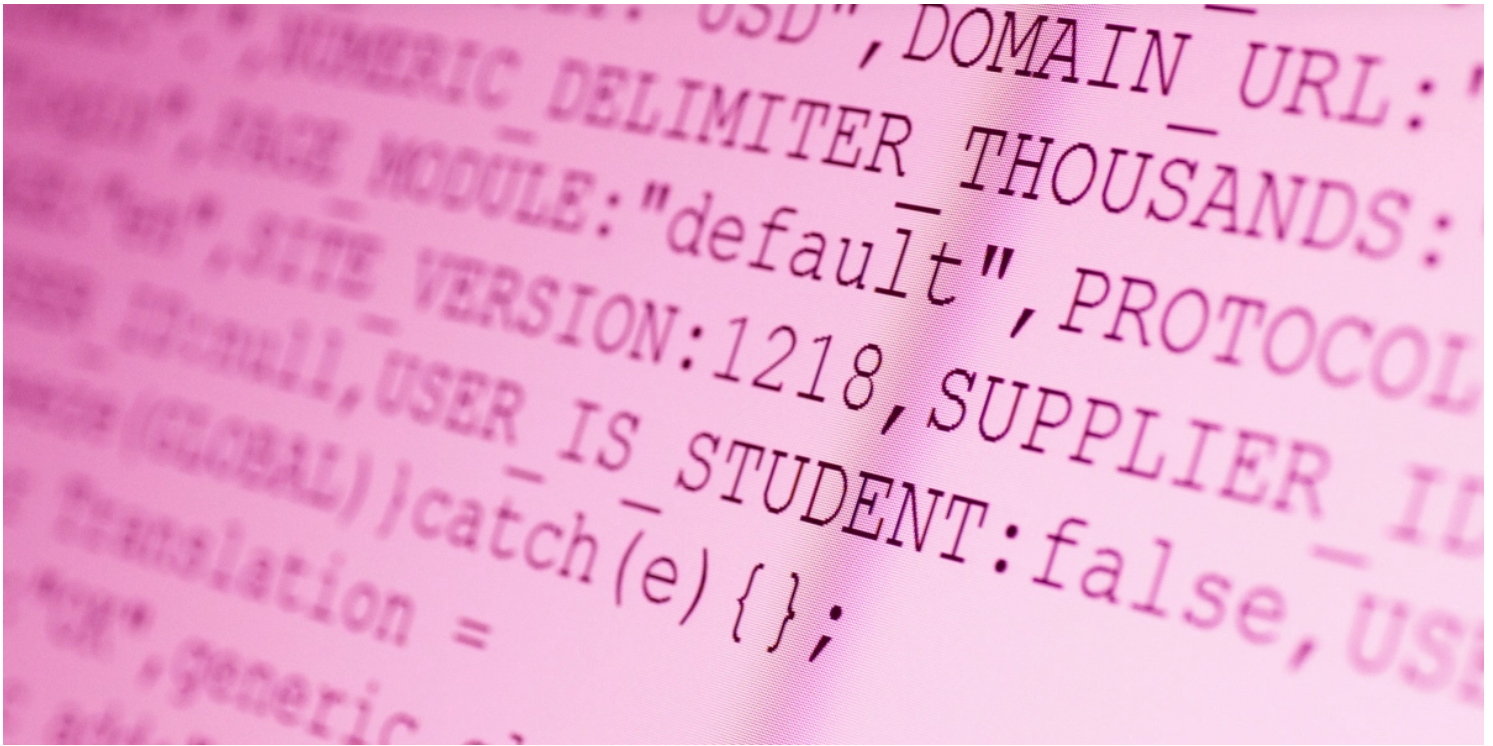
July 18, 2021 6:21 AM

Image Credit: anakaMaharageDharmasena/Getty

The *Transform Technology Summits start October 13th with Low-Code/No Code: Enabling Enterprise Agility.* [Register now!](#)

An emerging breed of tools are using machine learning and other methods to automate parts of the software development process. GitHub, for example, [launched such a tool last month](#) that suggests code while a programmer is developing it. Amazon has also created [CodeGuru](#), a tool to help automatically find performance bottlenecks in software. Facebook has [Aroma](#), which can also provide code-to-code recommendations. And my own team at Intel Labs has built a [tool](#) (currently only for our in-house use) that autonomously detects errors in code.

This kind of automated coding is called "machine programming." One of its most interesting capabilities is "code semantic similarity," which attempts to autonomously determine whether two code snippets show similar characteristics or achieve similar

"[big code](#) data" such as IBM/MIT's new [Project CodeNet](#) which includes approximately 14 million code samples, and and new machine learning algorithms.

By harnessing the power of code semantic similarity, the industry can develop automated systems to help CIOs ensure developer teams are maintaining the same level of productivity despite increased software and hardware complexity, all the while addressing the software developer talent shortage and combating burnout.

Enabling language-to-language translations

Code semantics similarity could also be used in tools that translate between programming languages (i.e., transpilers). Historically, software systems that convert a program's source code in one programming language to another was out of reach. However, recent advancements in transpilation could be critical for large, global organizations that have traditionally coded in more specialized legacy languages

Imagine a world where, instead of spending many years manually translating an entire organization's code bank from COBOL to Python, a machine programming system could do it all for you — in just a few days. The beginnings of such systems already exist and are even used in some tech companies today, such as Adobe. For example, Adobe Photoshop, as I understand it, is using verified lifting to convert C/C++ to Halide in its current version.

Code semantics similarity systems – such as [machine inferred code similarity](#) (MISIM) — will not only help an organization to update its *entire* code system; they will also open up the talent pool. Updating an organization's code base to a modern programming language from older legacy languages that are less understood by today's software developers will make recruiting easier as more developers are familiar with these newer languages (e.g., moving from FORTRAN to Python). CIOs might even see a reduction in coding errors because new-age languages tend to be easier to work with and handle much of the system complexity internally.

Elevating novice developers, helping to fill the developer gap

Code semantics similarity systems can also recommend code. GitHub's Co-Pilot, which I mentioned earlier, for example, is designed to learn what the intent of a piece of software is and then recommend improved (or more complete) versions to help the developer.

When fully realized, such code recommendation systems have the potential to raise the software quality and productivity of both novice and expert developers by providing them with improved alternatives. Ultimately, this will help CIOs and their IT departments keep up with software demands without hiring additional employees or spending money on new resources. The blue-sky vision of these recommendation systems is to improve the productivity of all developers.

Semantics similarity systems can also work in tandem with developers to autonomously

The bottom line

The landscape of software development is growing in complexity due to software and hardware heterogeneity. Development teams are also expected to produce software at an increasing pace. Machine programming may be the only fiscally viable way forward for CIOs and the software development they oversee. So this is the right time to begin testing out emerging machine programming tools and seeing how to best implement them in your organization.

Justin Gottschlich founded and leads the Machine Programming Research group at [Intel Labs](). He is also principal investigator and co-founder of the joint Intel/NSF CAPA research center. In academia, he has appointments as the industrial advisory board chair and executive director for the PRECISE Center at the University of Pennsylvania (Penn), and is also an adjunct assistant professor at Penn in the Computer and Information Science Department

VentureBeat

VentureBeat's mission is to be a digital town square for technical decision-makers to gain knowledge about transformative technology and transact. Our site delivers essential information on data technologies and strategies to guide you as you lead your organizations. We invite you to become a member of our community, to access:

- up-to-date information on the subjects of interest to you
- our newsletters
- gated thought-leader content and discounted access to our prized events, such as **Transform 2021: Learn More**
- networking features, and more

Become a member

Find your dream job on VentureBeat Careers

Bloomberg
London
View 10 Jobs
(https://jobs.venturebeat.com/company/bloomberg)

Northrop Grumman
Falls Church
View 7264 Jobs
(https://jobs.venturebeat.com/company/northrop-grumman-2)

CGI Group, Inc.
Amsterdam
View 1668 Jobs
(https://jobs.venturebeat.com/company/cgi-group-inc-1)

Checkout.com
United Kingdom
View 261 Jobs
(https://jobs.venturebeat.com/company/checkoutcom-1)

Transform 2021

Join us for the world's leading
event on applied AI for
enterprise business &
technology decision-makers,

Join forces with OHUB & VB to include & hire 1,000 BIPOC students at SXSW

Sponsor & hire

Learn More

VB Lab   Newsletters   Events   Special Issue

Product Comparisons   Jobs

About   Contact   Careers   Privacy Policy

Terms of Service   Do Not Sell My Personal Information