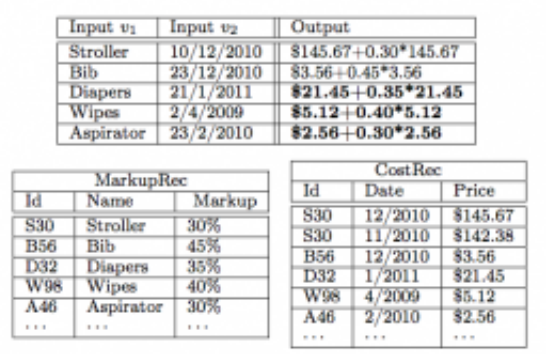
Excel programming for nonprogrammers

May 8 2012, by Larry Hardesty



Input v_1	Input v_2	Output
Stroller	10/12/2010	$145.67+0.30*145.67
Bib	23/12/2010	$3.56+0.45*3.56
Diapers	21/1/2011	**$21.45+0.35*21.45**
Wipes	2/4/2009	**$5.12+0.40*5.12**
Aspirator	23/2/2010	**$2.56+0.30*2.56**

MarkupRec		
Id	Name	Markup
S30	Stroller	30%
B56	Bib	45%
D32	Diapers	35%
W98	Wipes	40%
A46	Aspirator	30%
...	...	...

CostRec		
Id	Date	Price
S30	12/2010	$145.67
S30	11/2010	$142.38
B56	12/2010	$3.56
D32	1/2011	$21.45
W98	4/2009	$5.12
A46	2/2010	$2.56
...	...	...

In this illustration of the researchers' system in action, a user provides a pair of examples that correlate data — product names and dates — with an arithmetic expression. The system automatically determines the sources of the data — the tables below — and generalizes from the examples to additional expressions (in bold).

Microsoft's Visual Basic programming language lets Excel users customize their spreadsheets in all kinds of time-saving ways, but few people take advantage of it. Although designed to be intuitive and easy to use, Visual Basic can still be daunting to users with no previous programming experience.

As both an intern and a consultant at Microsoft, Rishabh Singh, a graduate student in the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT, has worked with Microsoft Research's Sumit Gulwani to develop a system that lets Excel users customize their spreadsheets simply by giving examples of how they want data

processed. Singh and Gulwani describe the work in a pair of papers they're presenting this summer at the Computer-Aided Verification and Very Large Databases conferences.

The system has two main components. The first, which Singh and Gulwani developed during Singh's first internship at Microsoft, can modify text strings on the basis of a few examples. For instance, if you had a spreadsheet in which dates were entered in the form "17 January 2009," you could simply type one or two examples of dates in the form "Jan. 17, 2009"; the system would automatically deduce that you want the order of the month and day reversed, the month truncated after the first three letters, and punctuation added in the appropriate places.

But over the last two years, Singh and Gulwani have greatly expanded the power of their system by allowing it to exploit tables that establish correlations between different types of data. Suppose that you had a column of data from an old database in which the date "January 17, 2009" was instead rendered in the form "011709." With the latest version of the system, you could create a 12-entry table correlating numbers with the names of months: "01" for January, "02" for February, and so on. Now, again on the basis of just a few examples, the system would learn to convert the numbers in the text strings to the names of the corresponding months.

Changing tables

Table-based correlations are particularly useful because, in practice, companies often have multiple databases or spreadsheets that contain different types of data about the same objects. For instance, a company might have Excel files from different retailers reporting sales figures and separate in-house documents recording R&D expenses for the same products. Using Singh and Gulwani's system, an Excel user could create a new file that compares aggregated sales figures with aggregated R&D

expenses, again just by providing a few examples. The user does not even need to specify which information came from which table: The system automatically finds correlations across different sources.

At the moment, Singh says, the system works only with text. So, for instance, it could learn to automatically produce text strings that represent mathematical relationships between data, but it couldn't evaluate them. That should be easy to remedy, however: Excel already has a "formula builder" function that converts strings such as "(323 + 73) / 9" into the corresponding mathematical expressions.

In the future, Singh and Gulwani's work could find its way into a commercial release of Excel. Indeed, the latest beta release of Excel includes a simpler example-based training system based on some of Gulwani's earlier work.

The blow-up

Technically, the chief challenge in designing the system was handling the explosion of possible interpretations for any group of examples. Suppose that you had a list of times in military format that you wanted to convert to conventional hour-and-minute format. Your first example might be converting "1515" to "3:15." But which 15 in the first string corresponds to the 15 in the second? It's even possible that the string "3:15" takes its 1 from the first 1 in "1515" and its 5 from the second 5. Similarly, the first 15 may correspond to the 3, but it's also possible that all the new strings are supposed to begin with 3's.

"Typically, we have millions of expressions that actually conform to a single example," Singh says. "Then we have multiple examples, and I'm going to intersect them to find common expressions that work for all of them." The trick, Singh explains, was to find a way to represent features shared by many expressions only once each. In experiments, Singh and

Gulwani found that they never needed more than three examples in order to train their system.

“If you look at the macros [small programs] that one would have to write in order to perform those text transformations manually, compared to the few demonstrations that you do as an end-user, it’s quite amazing how much programming you can avoid doing through this system,” says Rastislav Bodík, an associate professor of computer science at the University of California at Berkeley who specializes in automatic program synthesis.

Bodík says the researchers addressed two central technical problems. The first is that “you need to search humongous spaces of programs,” Bodík says, “so you need good algorithms for doing search in that space of programs that can prune effectively huge subspaces.” But the second problem “is the interaction with the user,” Bodík says. “It seems that they figured out a pretty nice system for how the user can convey to the synthesizer of programs what they have in mind without really having to know what’s going on underneath.”

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology

Citation: Excel programming for nonprogrammers (2012, May 8) retrieved 20 April 2024 from https://phys.org/news/2012-05-excel-nonprogrammers.html