The New York Times
SundayReview

GRAY MATTER

A Motherboard Walks Into a Bar ...

My computer's been trying to make me LOL all week. Listen to this: "Q: HOW MANY 13 EGGS DID LANCE ARMSTRONG BUY? A: A BIKER'S DOZEN."
How do you tell an algorithm you're not laughing *with* it, you're laughing *at* it?

David Rees

By ALEX STONE

Published: January 4, 2013

WHAT do you get when you cross a fragrance with an actor?

Answer: a smell Gibson.

Groan away, but you should know that this joke was written by a computer. "Smell Gibson" is the C.P.U. child of something called Standup (for System to Augment Non-Speakers' Dialogue Using Puns), a program that generates punning riddles to help kids with language disabilities increase their verbal skills.

Though it's not quite Louis C. K., the Standup program, engineered by a team of computer scientists in Scotland, is one of the more successful efforts to emerge from a branch of artificial intelligence known as computational humor, which seeks to model comedy using machines.

As verbal interaction between humans and computers becomes more prominent in daily life — from Siri, Apple's voice-activated assistant technology, to speech-based search engines to fully automated call centers — demand has grown for "social computers" that can communicate with humans in a natural way. Teaching computers to grapple with humor is a key part of this equation.

"Humor is everywhere in human life," says the Purdue computer scientist Julia M. Taylor, who helped organize the first-ever United States symposium on the artificial intelligence of humor, in November. If we want a computational system to communicate with human life, it needs to know how to be funny, she says.

As it turns out, this is one of the most challenging tasks in computer science. Like much of language, humor is loaded with abstraction and ambiguity. To understand it, computers need to contend with linguistic sleights like irony, sarcasm, metaphor, idiom and allegory — things that don't readily translate into ones and zeros.

On top of that, says Lawrence J. Mazlack of the University of Cincinnati, a seminal figure in the field of computational linguistics, humor is context-dependent: what's funny in one situation may not be funny in another. As an example, he cites Henny Youngman's signature line, "Take my wife — please," which came about by accident when an usher seating Youngman's wife mistook the comedian's request for a gag.

The cognitive processes that cause people to snicker at this sort of one-liner are only partly understood, which makes it all the more difficult for computers to mimic them. Unlike, say, chess, which is grounded in a fixed set of rules, there are no hard-and-fast formulas for comedy.

To get around that cognitive complexity, computational humor researchers have by and large taken a more concrete approach: focusing on simple linguistic relationships, like double meanings, rather than on trying to model the high-level mental mechanics that underlie humor.

Standup, for instance, writes jokes by searching through a "lexical database" (basically, a huge dictionary) for words that fit linguistic patterns found in puns — phonetic and semantic similarities, mostly — and comes up with doozies like: "What do you call a fish tank that has a horn? A goldfish bull."

Another tack has been to apply machine-learning algorithms, which crunch mountains of data to identify statistical features that can be used to classify text as funny or unfunny. This is more or less how spam filters work: they decide which messages to tag by analyzing billions of e-mails and compiling a database of red flags (like any urgent message from a deposed Nigerian prince).

Figuring out when a joke is a joke is where artificial intelligence researchers have made, perhaps, the most progress. For her Ph.D. dissertation, Dr. Taylor built a system that could identify children's jokes out of various selections of prose with remarkable accuracy. Not only that, but it could also explain why it found something funny, which suggests that on some level it "got" the jokes.

In a related experiment, the computer scientists Rada Mihalcea at the University of North Texas, Denton, and Carlo Strapparava, now at Fondazione Bruno Kessler in Italy, trained computers to separate humorous one-liners from nonhumorous sentences borrowed from Reuters headlines, proverbs and other texts. By analyzing the content and style of these sentences, the program was able to spot the jokes with an average accuracy of 87 percent.

Putting such research to good use, a pair of wags at the University of Washington last year taught a computer when to use the refrain "That's what she said" — theirs being one of the few academic papers to cite "The Office" among its references.

Some will surely wonder if the point of such research goes beyond devising software that can make the C++ set crack up at hackathons. Thankfully, it does. The goal of computational humor, and of computational linguistics as a whole, is to design machines akin to the shipboard computer on "Star Trek" — ones that can answer open-ended questions and carry on casual conversations with human beings, even William Shatner.

In the process, scientists hope to gain insights into the nature of humor: Why do we laugh at certain things and not at others? Why does anyone watch "Two and a Half Men"?

If computer humorists can answer any of these questions, we won't just get a deeper understanding of how language works but also, ultimately, what it means to be human.

Alex Stone is the author of "Fooling Houdini: Magicians, Mentalists, Math Geeks and the Hidden Powers of the Mind."