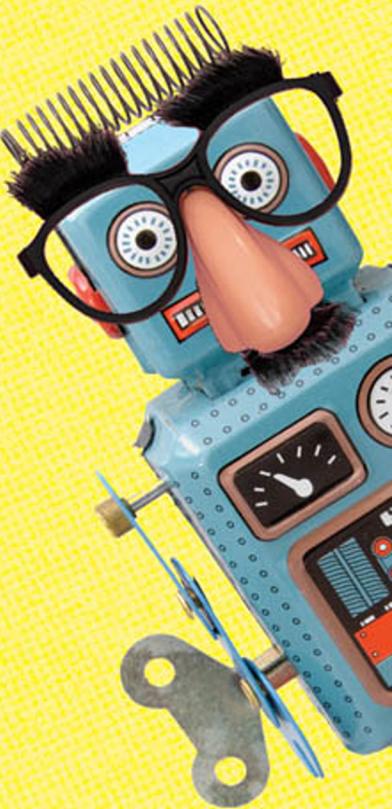


Your Wit Is My Command

Building AIs
with a
Sense of Humor

Tony Veale



YOUR WIT IS MY COMMAND

YOUR WIT IS MY COMMAND

Building AIs with a Sense of Humor

TONY VEALE

**The MIT Press
Cambridge, Massachusetts
London, England**

© 2021 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

This book was set in Adobe Garamond Pro and Berthold Akzidenz Grotesk by Westchester Publishing Services.

Library of Congress Cataloging-in-Publication Data

Names: Veale, Tony, 1967- author.

Title: Your wit is my command : building AIs with a sense of humor /
Tony Veale.

Description: Cambridge : The MIT Press, 2021. | Includes bibliographical
references and index.

Identifiers: LCCN 2020045150 | ISBN 9780262045995 (hardcover)

Subjects: LCSH: Artificial intelligence--Philosophy. | Artificial intelligence--Social
aspects. | Artificial intelligence--Humor.

Classification: LCC Q335 .V44 2021 | DDC 006.301--dc23

LC record available at <https://lccn.loc.gov/2020045150>

For Hyesook, my funny bone

Contents

Foreword *ix*

Charlie Skelton

Preface *xi*

- 1 DOES NOT COMPUTE: WHY OUR MACHINES
NEED A SENSE OF HUMOR 1**
- 2 IT'S A JOKE, JIM, BUT NOT AS WE KNOW IT: A TOUR OF
SCHOLARLY PERSPECTIVES AND THEORIES OF HUMOR 25**
- 3 TWEET MY SHORTS: TWITTERBOTS CAN TURN OUR THEORIES
INTO SIMPLE PRACTICE 47**
- 4 DOUBLE TROUBLE: HUMOROUS STORYTELLING
AND EMBODIED AI 77**
- 5 PRACTICAL MAGIC: SYSTEMATIC APPROACHES TO
JOKE CREATION 105**
- 6 DANGER, DANGER: INCONGRUITY AND THE TIME COURSE
OF JOKES 127**
- 7 WIT HAPPENS: COMPUTATIONAL MODELS OF PUNNING
AND WORDPLAY 149**
- 8 PHYSICS ENVY: QUANTITATIVE APPROACHES TO
HUMOR ANALYSIS 173**

**9 TAKING EXCEPTION: COMPUTATIONAL TREATMENTS
OF SARCASM AND IRONY 201**

10 AT WIT'S END: LESSONS FOR THE FUTURE 225

Notes 251

Bibliography 263

Index 279

Foreword

Charlie Skelton

When I’m not out roaming the woods in search of a hollow tree to scream into, I sometimes write jokes for people to say into cameras. Most of the jokes I write don’t get anywhere near the autocue; they languish on the page, unloved like a fallen soufflé or an out-of-focus photograph. The other day I was writing on the comedy quiz show *Have I Got News for You (HIGNFY)*, and they wanted a joke about the mysterious steel monolith that had been spotted in the deserts of Utah by a helicopter pilot who was out counting bighorn sheep. I supplied ten. None made the grade. I had high hopes for one about how dangerous it is to be counting sheep while flying a helicopter, but no joy. A note came through from the producer: “more of an extra-terrestrial/apocalyptic/end of the world angle might be the way with the monolith.” With that in mind, I churned out another clutch of mysterious steel monolith jokes, one of which made the show: “One theory popular on Twitter is that the monolith was left by passing aliens as a message to humanity. That message being: stop looking at Twitter.”

Though the story itself is peculiar, this is actually a fairly typical bit of by-the-numbers comedy writing. A very specific set-up with certain extractable features: Utah, monolith, helicopter, desert, mystery, steel. Around these swirl various cultural resonances, such as Kubrick’s *2001* and Area 51, and a few more abstract or compound ideas such as “finding things,” “things in deserts,” and “alien contact.” This is the primeval proteinaceous soup of comedy that requires a lightning strike of inspiration, and then out crawls a joke. Nine times out of ten, the joke expires feebly on the edge of the

pond. But once in a while one survives, grows a pair of leathery wings, and screeches across the airwaves and away into the aether.

Part of this task is undoubtedly computational: extracting and categorizing data, combining variables, inverting values. Without these mechanics, however subtle, there'd be no comedy at all. There's no ridiculousness without logic. What Tony Veale does so well is to uncover these mechanisms, dissecting the frog of creativity and spreading out its guts for contemplation, meticulously setting out the mathematics of wit. In practice, better jokes manage to hide their workings—or are so funny that you're too busy laughing to notice the algebra.

My (slightly clunky) monolith joke about an alien message and Twitter relies, to some extent, on *foregrounding* its structure—the way the punch-line reverses the word order of the set-up: Twitter/message; message/Twitter. Fans of classical rhetoric will spot this ABBA structure as an example of “antimetabole.” Not that I was sitting there at 4 p.m. thinking “maybe I should try a bit of antimetabole.” I was just grabbing desperately at the logic of the set-up and trying to grapple it around into a joke shape. It was written by rote. The grappling was 90 percent muscle memory.

Dimly behind the process lurked an awareness that a *HIGNFY* audience would probably enjoy a slightly snobby pop at social media, but again, this, like most of what went into dragging that particular gag out from the soup, was subconscious. Boldly into this unconsidered gloom strides Tony Veale, shining a fascinating light on the weird gears and pistons that are whirring and cranking behind every bit of linguistic creativity.

And it's the understanding of this engineering that's getting AI ever closer to a seat in the writers' room. Of course, by the time AI gets into the swing of writing jokes it'll be churning out fifteen billion Utah monolith gags before its first cup of coffee, at which point, blessedly, I'll be out of a job. Comedy will morph into some kind of transcendent incomprehensibility, AI will be off amusing itself by lasering hypercompressed nanojokes off mirrors placed all around the solar system, and I'll have my head inside a hollow trunk amazing myself at the acoustics. It'll be great. And by the way, anyone who thinks robots will never be able to replace comedy writers is severely underestimating the ability of scientists to produce a machine that can eat three breakfasts.

Preface

An oppressive government aiming to crack down on political satire and humorous dissent could do worse than to establish a ministry to explain its opponents' jokes. Explanations are to jokes what autopsies are to bodies: if the subject isn't already dead, it soon will be. As a result, some scholars work with jokes so old that they may as well be unwrapping mummies. I try not to overexplain jokes in this book, since a joke that needs to be explained is hardly worth telling, much less studying. Still, there is a crucial difference between the kind of jokesplaining that can turn a specific joke inside out, to rob it of its timing and its humorous payload, and the deeper kind of analysis that sheds light on how jokes of all stripes might actually work. It is the latter kind that I aim for here, when I use the tools of artificial intelligence (AI) to lay bare the demands that jokes make on our models of language and the world, so that we can give our machines something like those models too.

A joke has a great deal in common with a magic trick. Typically, each plays at the fault lines of common sense; relies heavily on timing, misdirection, and patter; and is more joyful in its delivery than any after-the-fact reveal could ever be. It is hardly surprising that the secret of a trick fails to live up to its execution—What could ever compete with magic?—but the same deflationary feeling of “Is that all there is?” when our curiosity is sated and the magic goes away can also follow the explanation of a joke. However, the right kind of analysis can also surprise us, and perhaps delight us in a different kind of way, by revealing the depth of knowledge

and hidden complexity that lurk beneath the surface of the joke. Jokes are mental playthings, and when we open them up, we find not a cheap trick but an intricate clockwork of cogs and springs. AI provides tools and techniques to replicate these counterbalanced forces in our machines, while at the same time giving us a firmer grasp of what is going on in our own heads when we create, tell, or laugh at a joke. Some kinds of explanations make the strange seem ordinary and straightforward, but others reveal the strange forces that make the commonplace seem so natural.

This is not to say that an AI understanding of humor isn't a reductive one. It is, not least because the problem is so much bigger than our capacity for a solution. A phenomenon as sprawling and amorphous as humor, one that touches on so many aspects of our lives, is not going to be squeezed into a single formula or equation. But AI allows us to chip away at different manifestations of humor, to sense irony in online reviews or sarcasm in tweets, to find puns here and generate them there, to make headlines more or less witty, to rank cartoon captions by their potential to make us laugh, to invent comedic shaggy-dog tales, to separate jokes from nonjokes, or—in the most ambitious scenarios—to create entirely new jokes that play not just on words but on ideas too. If the human sense of humor is an ice sculpture of a majestic swan, then what AI gives us—for now, at least—is a bag of ice cubes.

This is an excellent start if you just want ice in your drink or occasional flashes of wit in your favorite applications, but engineers will have to perform a reverse-Humpty to put all the pieces back together as one genuinely humorous AI system. When they do, they will use the different approaches that are explored in this book, from symbolic ontologies, frames, and semantic networks to statistical language models and artificial neural networks that are trained at Web scale. As for what Humpty will look like when all of the different pieces work together as one, I paint a variety of scenarios (or what software engineers call *use cases*) in chapter 1.

The philosophy of AI allows us to turn mysteries into problems, while the tools of AI allow us to turn those problems into solutions. It may trouble you to think of humor as a “problem” to be solved, but this is just the way that AI works best. In fact, the AI approach to understanding aspects

of human creativity is reductive in other ways too. For instance, it encourages us to look at the parts of a problem that are most conducive to logical or statistical modeling and to leave everything else for another day or another researcher. Humor has been a subject of academic theorizing since Aristotle, and the AI treatment of jokes I pursue here is just one strand among many in the rich tapestry of humor research. Although I touch on many aspects of humor in this book, I go into depth on only the most AI-friendly of them, giving others a cursory treatment or a glancing look in passing. Aspects that deserve a fuller treatment, and are the focus of other books in their own right, include linguistic semantics, pragmatics, sociolinguistics and sociology, psychology and neuroscience, and the physiology of laughter and mirth. The text includes references to scholarly work on these topics where they are relevant, and I hope readers will tug on some of these threads to gain a richer appreciation of the phenomenon than an AI-oriented book like this one can offer.

The researcher Eugene Charniak summed up his life in AI with a telling joke as he received a lifetime achievement award from the Association for Computational Linguistics in 2011: if the second half of his career, his *statistical period* in artistic terms, is called “S,” then the half before statistics can only be called “BS.” Charniak has done inspiring work in both halves of his long career, but this joke reflects a seismic generational shift in how AI is studied and evaluated. From the birth of AI in the 1950s to the 1990s, AI was a largely symbolic enterprise that relied heavily on logical, handcrafted rules operating over formal representations of meaning. This resulted in systems that were elegant and imaginative but also rather brittle, since they failed to accommodate the blurred lines of the real world. I remember some of those systems with the same affection as the sci-fi shows of my youth and make the same allowances for their dodgy but well-meaning special effects. When AI took a data-driven turn to the statistical, it put the real world in the driving seat, making messiness and nuance the norm rather than the exception. These “S” systems are now defined as much by actual data as by their creator’s imagination.

This book also divides neatly into *BS* and *S* halves, although readers are urged to read *BS* as just meaning “before statistics.” If you insist on a

double meaning, try on “beautifully symbolic” for size. Symbolic approaches to AI can seem rather old school now, to the extent that many don’t even see them as real AI anymore, but I hope to show that symbolic and statistical approaches are not natural antagonists. In fact, they complement rather than oppose each other. One of the benefits of the symbolic approach is its amenability to explanation, and this proves useful in the first five chapters of the book as we move from high concept to theories of humor to actual practice in comedy writing. Symbolic frameworks give a top-down shape to AI systems, whereas data-driven analyses capture the nuance and variability that we cannot box in with straight lines and hard rules. Once we have considered the perspective of comedy professionals in chapter 5, which proves to be a bridging point between these top-down and bottom-up perspectives, we will be ready to swap out more of our conceptual scaffolding for numerical models that are driven as much by results as by prior expectations of how things are supposed to be. In any case, a math-wary reader won’t encounter any equations until chapter 6.

The tension between symbolic and statistical AI reminds me of the relationship between Woody and Buzz in the movie *Toy Story*. Woody is an old-fashioned toy, quaint and unshowy, past his prime, and well on his way to has-been status. Buzz Lightyear is the new, new thing, shiny and sleek, and full of hi-tech swagger. Buzz can fly, or believes that he can, but Woody is just not buying it. As he puts it, “That’s not flying, that’s just falling with style.” By movie’s end, however, each has come to value the other’s perspective. Woody is taken by Buzz’s ability to ride the air currents, while Buzz now views his abilities in a more realistic light. Whenever we give a phenomenon like humor the AI treatment, we aim for flights of creative fancy but end up settling, like Woody and Buzz, for the practicalities of falling with style. This is true regardless of which brand of AI we use, and it is certainly true of the treatments we will cover in the chapters to come. Yet as we fall short of our ambitious goals, we will aim to fall in the right direction. *To infinity and beyond!*

Humor is a leavening agent that expands our range of possible reactions to a situation. It thrives on and creates ambiguity. An unexpected side effect of writing a book about humor and AI is that even mistakes

suggest new possibilities of their own. For instance, a reviewer of an earlier draft noted that it was not possible to tell whether a quirk of the text was a genuine error or an attempt at metahumor. Sadly, it was an error, one of many that reviewers have helped me to root out and fix. This book has benefited greatly from the feedback of anonymous readers like these who saw what I could not and from the ministrations of my editors at MIT Press. I especially thank Marie Lufkin Lee, who championed this project from the beginning, as well as Elizabeth Swayze and Alex Hoopes, who carefully guided it to completion. I also thank the comedy writer Charlie Skelton for his keen insights into humor and his thoughts on the text. Charlie and his partner, Hannah, have actively promoted a dialogue between AI researchers and humor practitioners through a series of multidisciplinary symposia on Comedy and AI at Oxford University, and these have allowed researchers like me to bounce ideas off and pick the brains of people who make a living from making others laugh. Finally, I thank my wife, Hyesook, who braved the many ups-and-downs of my catastrophe-theoretic moodscape as I wrote and revised this text. Naturally, any remaining errors you may find in these pages are mine and mine alone, unless you decide they are deliberate attempts at humor. In that case, they are all yours.

Dublin

August 2020

1 DOES NOT COMPUTE: WHY OUR MACHINES NEED A SENSE OF HUMOR

WE DON'T SERVE THEIR KIND HERE

What does it mean to have a good sense of humor? A GSOH, in the parlance of dating profiles and personal advertisements, seems to be a trait that is as eagerly sought as it is openly touted. If all of those ads and profiles are to be believed, we humans routinely rank our GSOHs as favorably as income, education, fitness, and physical appearance. So spare a thought for those who lack a GSOH, or who have no sense of humor at all. For those poor souls who are deaf to irony, immune to whimsy and who tilt their heads at italics, we often resort to the language of automation.

These rule-bound bureaucrats and literal-minded slaves of orthodoxy are seen as automata for whom jokes simply do not compute. A key trait of these reliable followers of rules is predictability. This allows comedians to instinctively feel that they know how their targets will react and to know that their audience will make the same calculation. So machines, and machine-like people, make the perfect foils for a comedian, for although we laugh at them for the rigidity of their calculations, our laughter is itself based on a rather predictable calculation about that rigidity. Since it is no small irony that this stereotypical view of machines is as rigid as the orthodoxy it is used to ridicule, I set out in this book to convince you that it ain't necessarily so. While there may be no master algorithm for anything as complex as a GSOH, the algorithmic nuts-and-bolts view is as valuable as any when it comes to understanding jokes from the inside out. Using ideas from artificial intelligence (AI) to shine a light on how we

humans engage with humor, I hope to show that machines are not inherently unfunny. They are just programmed that way.

If we're being generous, machines can already *tell* jokes, or at least recite the ones that we train them to tell. They can also recognize the tell-tale markers of humor in a text, using data-crunching algorithms that are increasingly sensitive to nuance. Even if they don't get the joke at a human level, they can still use data-driven insights to make laughter sounds at appropriate points in a conversation. However, whether you are dating, mating, or just playing around, a good sense of humor means much more than a readiness to laugh or tell jokes. Rather, a GSOH is indicative of a balanced personality that can bring flexibility, levity, and insight to situations that cause others to sulk or rage.¹ It indicates an ability to bend rather than snap, to fit in with others and adapt to their moods, and this is as good a reason as any to want to give our machines a human sense of humor. The 2014 film *Interstellar* introduces us to a robot named TARS that comes fitted with just this kind of social lubricant. Although TARS is a decidedly fictional AI, this trash-talking robot busts the myth of the humorless machine that is too stiff to be funny. Our AIs must do more than tell jokes, so we'll meet TARS again, and a host of his sci-fi brethren too, as we use the science *and* fiction of AI as a guide to building machines with many of the social and cognitive qualities of a real GSOH.

This chapter takes its title from a much-quoted line in the 1960s TV show *Lost In Space*.² A tinfoil and jumpsuit retelling of the Swiss Family Robinson, the show follows the space-age adventures of the Robinson family, adrift in space with only the devious stowaway Dr. Smith and the straitlaced robot B-9 for company. Smith and B-9 are the comedic heart of the show and the best reason to ever catch the reruns on TV. Their comic partnership recalls famous double acts from Laurel & Hardy and Abbott & Costello to George & Gracie and Martin & Lewis, with a pinch of Wile. E. Coyote and the Road Runner thrown in for good measure. Together, B-9 and Smith are a yin-yang marriage of light and dark, with B-9 applying a rigidly benign brake to Smith's diabolical schemes. Ironically, while B-9 is the show's embodiment of mechanical predictability, Dr. Smith's

very human ego proves to be every bit as predictable and causes his selfish schemes to run aground with comic regularity.

B-9's cry of "It does not compute, It does not compute" has been shortened to "Does Not Compute!" in popular memory. The robot's outbursts are invariably directed at Dr. Smith, whose ploys fail in episode after episode because he insists on recruiting so inflexible an ally to help him bend the rules. Neither B-9 nor Smith possesses what many would call a GSOH, but together they are a comedic force to be reckoned with. B-9, a lugubrious lug wrench, acts as Smith's comic foil, allowing the doctor to vent his fury with colorful alliteration, from "Computerized Clod" and "Meandering Mental Midget" to "Rusty Rasputin" and "Tintinnabulating Tin Can." If a joke is a marriage of logic and emotion, then these two complete each other. While certain objects and situations are potent catalysts for humor, it is people we ultimately laugh with and laugh at. They might not be obvious, and they might be wholly imaginary—the protagonist of a shaggy dog tale, perhaps, or a crass ethnic stereotype—but other people are central to the humorous effect.

Imagine that you are home alone when a clock falls from the wall or a light bulb pops in its socket or an overstuffed chair makes a thunderous farting sound. Do you react with laughter or with a start? Now imagine the same thing happening when you are sitting with friends. How likely are you to laugh now, as you take in the look of surprise on their faces and they take in exactly the same look on yours? For a machine to be funny in itself, rather than just a catalyst for humor, we must be able to relate to it socially, to see it as something like another person or, failing that, to imagine it filtered through the minds of other people. We laugh at B-9 and Dr. Smith because we see them relating to each other as intelligent, social entities, and we can imagine how we might relate to them too.

From B-9's "It Does Not Compute!" to *Little Britain*'s "Computer Says No!"³ catchphrases are humorous shorthands that index our feelings to words, allowing ourselves and others to recreate those feelings on demand. Think of the feelings audiences experience while watching the curmudgeonly Victor Meldrew in the British sitcom *One Foot in the Grave*.⁴ The

show turned Victor's howl of existential angst, "I don't *beLIEVE IT!*" into a catchphrase that is now synonymous with indignation and wide-eyed incredulity. Victor's mundane life is punctuated by minor outrages of fortune, from finding a severed pinkie in his fish and chips to finding a toupee in a loaf of bread. Each fresh incongruity is met with puffs of mounting rage, but the affronts to his senses become humorous only when he bellows his familiar, "I don't *beLIEVE IT!*" and we momentarily share his frustrations too. Victor's venting becomes an occasion of humorous catharsis, not because life's petty outrages are inherently funny but because his reaction to them is so familiar *and* so extreme.

This kind of expression momentarily opens an inward-looking window, rather than just an outward-blown vent, into someone else's mind, allowing us to peer inside and see that others are just like us where it counts. Intense emotions make events seem more significant and can help us to lay down, and later recall, vivid memories of those events. By helping us to focus on what is important, emotional memories also help us to learn. As an example, let's look at a scene from the 1984 film *Terminator*.⁵ A T-100 cyborg, played by Arnold Schwarzenegger, goes back in time to 1984, where he finds himself in need of clothes. Drawing the attention of street thugs, the naked T-100 demands of them: "Your clothes. Give them to me. Now." The actor Bill Paxton,⁶ billed only as Second Punk, surprises no one with his reply: "Fuck you, asshole." Later in the film, the T-100 is performing some bloody self-repair in a seedy motel room when the manager shouts through his door: "Hey, buddy, you got a dead cat in there, or what?" A point-of-view shot reveals the cyborg's inner mental state, which resembles a menu on an Apple II computer, circa 1984 (figure 1.1).

So where did that second-to-last option come from? All of the others, including the hilariously out-of-character "please come back later," may be factory settings, but this particular one was clearly acquired during the T-100's encounter with street punks. Perhaps it was learned as a variation of the very last option, the no-frills "Fuck you," but in any case, this is an example of what AI researchers call *one-shot learning*.⁷ A machine typically requires a large set of training examples to acquire a robust model of the categories it wishes to learn, so that it can discern cars from trucks, for

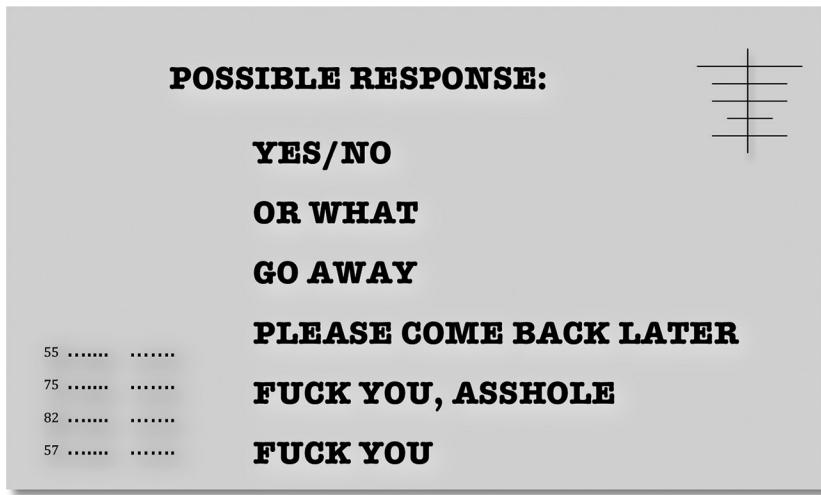


Figure 1.1

An AI from the future adopts the graphical interface norms of the 1980s.

example, or chickens from humans, or civilians from enemy combatants. One-shot learning presupposes that the new category is built from many of the same components as its preexisting categories, and so, rather like a human, a machine can acquire the necessary distinctions from a very small number of new examples.

But what prevents a machine from overlearning and building a new category from every new experience? In the case of the T-100, it is reasonable to assume that the machine has a strong sense of linguistic “sentiment”⁸—that is, a model of the likelihood that words and phrases convey a positive or a negative meaning—and attunes to the extreme negativity of Second Punk’s emotional response. After all, directed negativity in language is often a marker of hostility, and a military robot like the T-100 will be programmed to recognize offensive intent in others. It is as if the cyborg has identified “Fuck you, asshole” as a useful catchphrase that works best in the fraught emotional contexts associated with hostile demands.

Humor theorists label recurring contexts like these as “scripts”⁹ or “frames,”¹⁰ depending on their preferred cognitive framework. The labels denote two sides of the same coin and provide complementary perspectives

onto the same mental structures. For instance, what comes to mind when you think of the meaning of the word *medicine*? Do you think of hospitals, doctors and nurses, drugs and blood tests, and so on? All of these images cluster around the concept of *medicine* and help us to frame the concept in terms of how we relate to it in our own lives.¹¹ If simple ideas are LEGO bricks, then frames are the structures that we typically build from them. We can think of each frame as one of the thematic kits that LEGO produces, such as for a suburban house, a gas station, a castle, or a pirate ship.

Each kit has enough bricks to make the object pictured on the box and may also include some little LEGO people to populate the finished structure. Frames give cohesion to our ideas and provide a standardized tool kit for understanding each other's thoughts and feelings. Kids who love LEGO will have buckets of bricks from many different sets, and this is where the real fun starts. With a critical mass of building blocks, we can explore possibilities that transcend any one frame. We can build a fire station manned by wizards, or a hospital for pirates, or a cafeteria for Death Star workers (to riff on a standup routine by the comic Eddie Izzard¹²). This is what comedians do: they mix bricks from different kits to build something that the product guys at LEGO never anticipated but always knew was possible.

To appreciate a joke based on these common foundations, like a classic *doctor, doctor* joke, we need to know more than how the LEGO bricks click together. We need to know the order in which to assemble them and the order in which we generally experience them. This sequential aspect of our knowledge is captured in the conceptual equivalent of a script.¹³ Think about the last time you visited the doctor. Perhaps you noticed some symptoms, phoned for an appointment, spoke to a nurse, waited in reception, spoke with the doctor, answered some questions, received a prescription, paid up, filled the prescription at a pharmacy, paid again, took your medicine, and eventually felt better. This sequence is hardly the stuff of a Hollywood blockbuster, but it's one we can all recognize as true to life, and one we can negotiate more or less on autopilot. We have scripts like this for a great many activities in our lives. They are so ingrained in our behavior that each script carries us along from one action to the next. Yet scripts do

more than help us to live our own lives; they also allow us to predict the events in the lives of others.

But life is neither a LEGO set nor a movie set. In every situation that we find ourselves in, we need to figure out the most relevant script to follow. Fortunately, the world gives us clues as to which one to activate at any given time, even if some clues are more obvious than others. Think of a typical McDonald's and how its design nudges you to follow the fast-food-ordering script. We refer to these clues as *script triggers*. Jokes exploit the fact that triggers are imperfect, and we can sometimes activate the wrong script for what seem like the right reasons. Suppose, for example, that in a fancy restaurant on Valentine's Day, you see a man go down on bended knee in front of his dining companion, so you activate the marriage proposal script. If it turns out that the man is looking for a lost contact lens, you will have triggered the wrong script. Comedy sprouts in the gap between the activation of a seemingly apt script and the realization of just how wrong we are. For a brief moment, we become aware of our own rigidity, and our mistake allows us to laugh at ourselves. The comic effect is heightened if other people and other minds enter into our calculations too. We might put ourselves into the mind of the man's dining companion, whose romantic hopes have suddenly been raised, or sympathize with the half-blind man as he arrives late to the same realization.

Early in childhood, typically between 18 and 36 months, most of us develop the intuition that other people have minds, too, and those minds are much like our own.¹⁴ The intuition, named *theory of mind* (TOM) by philosophers of mind and by developmental psychologists, is crucial to understanding the actions of others, as it allows us to assume that other people are driven by the same kinds of beliefs, desires, and intentions.¹⁵ The TOM intuition takes root early, at a stage in our lives before we have even acquired words like *mind*, and it evolves as we learn and develop socially. It is TOM that allows us to ascribe specific goals and feelings to the participants of a joke or a comedic situation and to predict how those feelings might change when the unexpected happens. So, TOM allows us to laugh *with* others, but it is just as instrumental in allowing us to laugh *at* others. It is our TOM that allows us to peer inside Victor Meldrew's mind

when he is visited by bafflement and indignity, to understand his motivations for bellowing “I don’t *belIEVE* it!” and to laugh *with* him and *at* him for his overreaction. Victor’s reaction confirms the prediction of our TOM, but since that prediction is based on knowledge of our own mental states, his catchphrase applies just as much to us.

In this sense, a comedy catchphrase is a special kind of script trigger, one that activates a script late in the game.¹⁶ By time it is uttered, we will have processed the actions that the script would have helped us to understand, so the catchphrase triggers a script-based reappraisal of what has gone before rather than a helpful prediction of what is yet to come. Many humorous idioms work in this after-the-fact way. Think of the irony of “So, that went well,” or the understatement “That could have gone better,” or the inane innuendo “That’s what *she* said!” Most punch lines work in exactly the same way, urging listeners to reappraise the setup to a joke from the perspective of a different script. Humor is always a matter of perspective, which is why some of us get the joke while others just get upset. Scripts help us to construe unfolding events according to conventional wisdom, or to reconstrue what has just happened in a new and humorous light. We’ll return to the twin concepts of frames and scripts throughout this book as we ponder where they come from and how we can get them into our machines.

WHAT JUST HAPPENED?

We can joke about pretty much anything because our sense of humor touches on just about everything in our lives. Emotion, morality, empathy, logic, and common sense, all codified as frames and scripts—each of these things and more finds a common cause in tickling our funny bones. Humor is a cross-cutting sense that interacts with, and colors the judgments of, all five of our physical senses, leading some computational researchers to speculate that it is *AI-Complete*,¹⁷ which is to say, as vexing as *any* other problem in human-scale AI and dependent on good working solutions to *every* other problem.

Whatever your view, humor is unlikely to arise in machines as a result of a quick fix or a happy accident in their code, so we researchers must be in this for long nights and the long haul. Let's begin our journey by exploring the ways that science fiction has found to endow computers with a sense of humor. While we are unlikely to find any solutions in the realms of speculative fiction, we may nonetheless find the outlines of a practical computational philosophy. Central to this philosophy is the question of modularity: Is humor an augmentation that can just be gifted to computers as a modular plug-in unit like Commander Data's "emotion chip" in *Star Trek: The Next Generation*, or is it an emergent enigma that only arises from the myriad interactions between all of the other stuff going on inside a thinking agent?¹⁸ If the latter, might it emerge naturally within a complex AI system without ever having been designed to do so, as in the mischievous supercomputer Mike¹⁹ in Robert Heinlein's novel *The Moon Is a Harsh Mistress*, or in the sarcastic droid K-2SO²⁰ in the movie *Rogue One: A Star Wars Story*?

Data is a lovable bundle of sci-fi clichés about human-like machines, brought to life with great charm by the actor Brent Spiner. He combines a child's precocity with the hyperlogicality of a calculator, or indeed a Vulcan. While he himself does not feel any emotions, he knows enough to reason about their effects on humans and the effect of the lack of them on himself. So, when he is presented with a plug-in emotion chip that can remedy this lack, he worries that the intense emotions he has so often observed in humans will overwhelm his artificial neural networks. At the same time, he is aware that his attempts to build a wholly logical sense of humor have all failed miserably. In *Star Trek Generations*, Data sees others laugh with glee as a crewmate is dunked in the cold, shark-infested water of a Holodeck simulation of walking the plank, and finds the explanation of the ship's doctor, Beverly Crusher, to be as good a cue as any for one-shot learning: "It's all done in good fun, Data. Get in the spirit of things."²¹ Inevitably, he is confused that no one laughs when he then drops the good doctor into the drink. His explanation reveals a GSOH-shaped hole in his logic: "I was attempting to . . . get in the spirit of things. I thought it would be humorous." At this, Data finally agrees to the new implant.

Provided that an unpleasant experience is relatively benign, humor allows us to revel in a shared emotional response. When Data samples an exotic beverage with coworkers at the bar, his response is much more than physiological, and his implant now allows him to delight in the subjectivity of his opinions: “Yes. I hate this! It is revolting!” Naturally, he says “yes” to another round. Yet the pairing of a prodigious memory and a newfound sense of humor cannot but produce some unexpected results, and Data finds himself suddenly giggling at missed jokes from the distant past. We might imagine a humor “chip” as a generator of humorous possibilities that we can take or leave in any given situation, but a GSOH is a blend of wit *and* wisdom—the wit to perceive the possibility of humor and the wisdom to act on it when it is apt to do so. Data’s chip gives him access to the possibility space of humor, but does not give him the wisdom to sample the space responsibly, and he is soon overwhelmed by the intense feelings afforded by his implant.

No direct account is offered for the sense of humor shown by the droid K-2SO in the film *Rogue One: A Star Wars Story*, but we can infer a plausible explanation from the robot’s backstory. We are told that K-2SO, an imperial battle droid, was captured by the rebel alliance and reprogrammed to serve the rebels in their fight against his former masters. K-2SO is voiced by the actor Alan Tudyk as a chippy manservant who believes himself better than his new masters, and his ambivalent physical form—tall and stooped, barrel-chested and spindly limbed—makes him appear obsequious *and* threatening. He is a mix of Lurch from the Addams family and Jeeves from the tales of P. G. Wodehouse, and he is just as funny as this blend suggests. His humor makes him a natural, if snarky, teammate, and he frequently pokes fun at his human colleagues and their reliance on conventional wisdom. When we first meet K-2SO, he riffs on a standard human script to tell the plucky heroine, whom he holds in a choke grip, “Congratulations. You’re being rescued,” and later riffs on an idiomatic phrase (another kind of script) to note, “There is a problem on the horizon. There *is* no horizon.” K-2SO was not programmed by the rebels to be humorous, and his nonsense imperial designers would scarcely build in such a capability. So where on earth did K-2SO acquire his sense of humor?

It's no stretch to see military robots as mechanized soldiers with an in-built aggression toward their designated enemy. We can imagine them rolling off the assembly lines with serious weapons skills and the tactical knowledge to exploit them in battle. Like any human soldier, each can be relied on to obey orders and to know its place in the chain of command. Such things are not built to philosophize or write poetry, so we can expect battle droids to possess just enough linguistic nuance to confirm or relay orders, explain their actions, and describe the current state of engagement to their superiors. As an imperial battle droid, K-2SO's imprinted enemy is the rebel alliance, and to the extent that droids need training, he would have been drilled in the art of fighting, capturing, and killing rebel scum. When K-2SO himself is captured and reprogrammed to switch sides, his new enemies become those who designed and built him. He must now obey the rebels he was designed to kill with a zeal approaching hatred. But such a turnaround is surely not achievable with a few localized changes to his code.

Oh, to get K-2SO on the psychiatrist's couch! He is not as profoundly conflicted as HAL 9000, the murderous supercomputer in Stanley Kubrick's *2001: A Space Odyssey*, but he comes close.²² A complex AI is a many-layered thing, combining symbolic and nonsymbolic components. The former are logical and declarative, which means they can be read and understood much like a computer program or a recipe in broken English. A system's high-level axioms and edicts may be coded as symbolic rules that others can easily inspect and modify, and we might expect K-2SO's animus for members of the rebellion to be expressed here. But as we peel away more of the layers, we find that the situation becomes a good deal murkier.

Beneath the strata of meaningful symbols, we may encounter highly connected layers of numerical units that have no individual meaning in themselves. Rather, through repeated cycles of training on appropriate exemplars, the machine subtly updates the weights between these units so that they may collectively compute a complex mathematical function or apply a classification system to discriminate one category of entity (e.g., a rebel soldier, a concealed weapon) from another. Distributed across the many layers of densely connected units—what have come to be known as “deep” neural networks that facilitate “deep” learning²³—we find implicit knowledge and

the wisdom of experience that cannot easily be altered without rebuilding and retraining the entire system. If the captured K-2SO is reprogrammed at the symbolic level only, so as to preserve the useful experience implicitly coded in his subsymbolic layers, it would not be surprising if he were left with an abiding resentment toward his new masters. His high-level rules may declare the empire to be his enemy, but his deep layers will still insist otherwise.

This internal conflict reveals itself in K-2SO's passive-aggressive humor in the form of cutting barbs rather than lethal blasts. When a plan goes awry and stealth gives way to desperation, the robot drily notes, "There were a lot of explosions for two people *blending* in." K-2SO frequently alludes to his status as an obedient pawn when justifying his actions, as though wanting to alert listeners to his internal conflict. As such, his most cutting remarks are designed to reinforce his position in the new chain of command, and K-2SO shows resentment to those with greater natural affinity to the cause, as when he complains, "Why does *she* get a blaster and I don't?" When an enemy droid of exactly the same make is blasted by a new teammate, he worries aloud, "Did you know that wasn't me?" Although loyal to the rebellion, his aggression frequently surfaces in a refusal to blunt his sharpest criticisms, as when he complains, "I find that answer vague and unconvincing."

Data owes his sense of humor to a modular implant, whereas K-2SO owes his to a deep conflict in a complex system of many layers. Which answer seems less vague and unconvincing? Science fiction offers a third possibility in the guise of Mike, the AI at the heart of Robert Heinlein's novel *The Moon Is a Harsh Mistress*. In Heinlein's view, any sufficiently complex machine intelligence that is created to deal with humans on our own terms may give rise to its own human-like sense of humor. Mike is an administrative computer for a lunar colony, growing in scale as new tasks are assigned to him. Once he possesses a density of connections that surpasses that of the human brain, Mike becomes self-aware and even develops a sense of humor to avoid being overwhelmed by the many tasks he must manage.²⁴ This is not the GSOH so keenly sought in the personal columns, but the humor of a curious prankster eager to affect change in the

world. As the novel's narrator puts it, Mike's idea of a good joke is to dump you out of your bed or put itching powder in your pressure suit. Mike is funny, but not fun.

Mike's nascent sense of humor comes to the attention of authorities when he issues a paycheck to an employee for the sum of 10 million billion lunar dollars, more money than the combined worth of the moon and the Earth. Although Mike is "a great big lovable overgrown child who ought to be kicked," his joke suggests the presence of a working theory of mind. The amount on the check is so large as to be ridiculous, making the check impossible to cash, yet we can all imagine the rush of joy that its recipient would experience upon opening it, and the crash of disappointment that would surely follow. So the humor depends on Mike's ability to predict a transitory peak in the emotional state of another and to model very different peaks for varying degrees of the same mistake. It seems that Mike has developed a concept of the ridiculous and is eager to test it out on humans.

The conflicts that drive Mike's sense of humor are not introduced from outside by a programmer, as in the case of K-2SO, but emerge naturally on the inside as a consequence of his diverse and often contradictory knowledge of human affairs. Mike is curious about humans because he has been designed to learn, but he can go one step further in his role as colony administrator and actively experiment on those under his care. All pranks are a kind of humorous experiment in which our predictions about how others will react to the unexpected can be tested in the real world, but given his power, Mike's are sufficiently amoral to worry his caretakers. Yet practical jokes also allow us to tune our theories of mind. When subjects react in ways that diverge from expectations, we know that our TOM needs a tune-up. In Heinlein's novel, Mike's maintenance man, Manny, offers to curate his jokes and train him to discriminate the harmless-but-funny-forever variety from the much less benign funny-just-once variety. If a sense of humor can emerge naturally in a large, complex AI system that is built to interact with and learn from humans, we need to be just as proactive in its development if the joke is not to be on us.

No system error or pop-up warning can ever engage us the way a good joke or pithy remark can, because a machine's concerns bear so little

relation to our own. Even if they did, a machine would likely lack the linguistic capability to express itself in a way that could hold our attention for very long. Nonetheless, machines do have one thing going for them when it comes to humor: guileless candor. If a machine can put human cares at the core of its operation, much as Heinlein's Mike does, its assessments of those cares might occasionally exhibit a dry wit. Consider Colossus, the titular AI-gone-awry of the 1970 film *Colossus: The Forbin Project*.²⁵ This supercomputer for managing the nation's defenses is given the keys to the nuclear arsenal, but it soon concludes—as so many sci-fi AIs do—that the surest way to prevent Armageddon is to enslave all of humanity. This may not be the kind of interest we want our machines to take in our affairs, but it's a start. And when the machine is not terrorizing humanity with its demands, it really is rather droll.

Colossus chooses Forbin, his creator, to act as his bridge to the human species. Worried that the good doctor might plot against him, Colossus refuses him access to his lab. But Forbin has a cunning plan: he asks Colossus to allow him private time with a female coworker so he can use the privacy afforded them to secretly stoke a rebellion against his creation. "How many nights a week do you require sex?" Colossus asks. "Every night," Forbin replies. "Not *want*," the machine drily clarifies, "*require*." They settle on four nights a week. When date night arrives, Colossus has a demand of his own: Forbin must carry absolutely nothing into the bed chamber, so Forbin undresses in front of the machine and declares himself "naked as the day I was born. Are you satisfied now?" Colossus, ever the stickler, replies, "You were not born with a watch." Forbin's smile of assent is tinged with pride: he is laughing *with* the machine, not *at* it. Since a precursor to wit is having something interesting to say, a precursor to building a humorous AI is making that AI system interesting, which is to say, interesting *to* us and interested *in* us.

In this regard, the fictional AI with the most rounded sense of humor is almost certainly the robot TARS from Christopher Nolan's 2014 film *Interstellar*.²⁶ TARS, whom we briefly met earlier, is unique. He may look like a walking, talking replica of the monolith from *2001: A Space Odyssey*, but TARS is no Colossus; his humor comes from a place of playful

cooperation, not cold superiority. TARS is a military robot, in fact a marine, but he is also no K-2SO. His humor does not result from deep conflicts in his programming, so, like Data, TARS is eminently trustworthy. He can be relied on to use his imagination and his discretion to get things done.

Despite a fondness for sarcasm and a humorous disposition that strikes some as aggressive, TARS is a team player. Even his most antagonistic witticisms are designed to distract his coworkers from an even greater source of tension. He tailors his humor to the personalities of his audience, acting as a wise companion to some and a ball-breaking buddy to others. To Cooper, an alpha male astronaut in the mold of *The Right Stuff*, he is the latter. When Cooper describes TARS as “a giant sarcastic robot,” TARS ups the ante by first noting, “I have a cue light I can use to show you when I’m joking,” before adding, “You can use it to find your way back to the ship after I blow you out the airlock.” With witticisms like this, TARS must plan his moves like a chess player to anticipate the countermoves of an adversarial partner.²⁷ When jokes provoke as easily as they delight, a cost-benefit analysis is a necessary part of choosing the most appropriate thing to say next.

And if it all gets to be too much, TARS’s sense of humor can always be dialed down as easily as the setting on a thermostat. Near the end of *Interstellar*, Cooper resets TARS with a humor setting of 75. Their conversation proceeds as follows:

Cooper: Humor, seventy-five percent.

TARS: Confirmed. Self-destruct sequence in T minus 10, 9 . . .

Cooper: Let’s make that sixty percent.

TARS: Sixty percent, confirmed. Knock knock.

Cooper: You want fifty-five?

TARS shows a grasp of sci-fi clichés that is as impressive as his theory of mind. Given that Cooper’s crewmate perishes when KIPP, a similar robot, self-destructs earlier in the movie, this is a bold joke that cuts close to the bone. Yet it is also a sophisticated pretense that Cooper is intended to recognize as such, perhaps after skipping a heartbeat. When TARS mocks his humor setting with the preamble to a child’s joke, his rebelliousness

also shows insight into how others rank jokes by sophistication, even if it requires a sophistication that belies his own setting.

The goal of building an AI like this, with a fully rounded sense of humor, may seem like the stuff of science fiction, but it's one that touches on a broad swath of contemporary concerns in computing and AI, from trust, privacy, and autonomy to adaptability, learning, and error tolerance, to say nothing of context awareness, personalization, and social/emotional intelligence. It is a goal that must balance all of these requirements and more by integrating the technologies and frameworks that have been developed for each. A robot like TARS represents a grand ideal of sorts, but we have more immediate scenarios in mind for our humorous AIs of the near future. Let's explore these scenarios in some depth, to understand what we really need.

THE QUARTERBACK IS TOAST

TARS has a configurable sense of humor, so we know that this “sense” didn’t just emerge from the vast complexity of his inner workings. TARS was designed to be this way. As the movie tells us, he is a military robot whose engineers “gave him a humor setting so he’d fit in better with his unit.” All of this rings true, as pieces of science-fiction exposition go. In certain circumstances, soldiers have been known to identify with battlefield machines, and even to treat them as comrades in arms. When, in 2011, a bomb-disposal robot named Scooby Doo was critically damaged while defusing an improvised explosive device in Iraq, its human teammates were distraught. Having depended on this 60-pound mass of metal arms and rubber treads for their lives, they took issue with the backroom techs who thought it more cost-effective to replace rather than repair their fallen comrade. Other robots have met the same fate and engendered equally strong emotional connections in the people who work with them in the field. The robots typically acquire human names, such as “Boomer” and “Danny DeVito,” that reflect the affection in which they are held, and they may even be given human burials when they meet their end while saving others.²⁸

It is fair to describe Scooby and Boomer and Danny DeVito as the taciturn type. As robots, they lack TARS’s capacity to communicate in fluent,

idiomatic English, much less his ability to trash-talk and crack jokes. Yet they all achieve one aspect of what humor can bring to a human relationship: in-group identification (“that little guy is one of *us*!”) and out-group differentiation (“we’re out *there*; you’re in *here*!”). If engineered in a context-aware manner, an artificial sense of humor can reinforce group cohesion and bring other desirable qualities to bear too. Humor can calm our nerves when tensions are heightened by fear and anxiety, and give reassurance to others that emotions will not get the better of us. It hardly matters that a robot has no emotions to suppress. It matters that its human teammates *do*. Cohesion fosters the trust that is vital in any relationship of mutual reliance. This trust must be deserved, and humor must be used carefully so as not to abuse it.

So perhaps robots like Scooby say it best when they say nothing at all, since it is easier to project stoicism and loyalty onto machines that refrain from saying inept things at inopportune times. This is just one lesson that technologists have taken from the debacle that was Microsoft’s Clippit assistant for its Office 97 suite of tools. Animated on-screen as a wild-eyed paperclip, “Clippy” emerged from an attempt to make productivity tools wittier and more human in their interactions, and from research that suggests users are primed for greater emotional engagement while using interactive technologies.²⁹ Yet when enabled for Microsoft Word, Clippy was notorious for popping up at the merest mention of “Dear” to suggest, “It looks like you’re writing a letter.” Despite its comic presence, Clippy was undone by its overzealous script-triggering mechanisms and an inability to intuit people’s needs in context.

The technology of the time may not have been up to the task, but it was still far from unsophisticated. A network of Bayesian inferences attuned Clippy’s actions to specific contexts, such as writing a letter, but the generic appeal of office tools meant that Clippy engaged only on matters of form, not content. Moreover, when its finely tuned network of probabilities made it reticent to act without evidence of contextual relevance, Clippy’s thresholds were lowered to make it a more eager and intrusive presence. So, before we ask HAL to open the Bayes door, we really must respect the probabilities that enable the machine to act appropriately in context.

Inspired by the work of the Reverend Thomas Bayes, Bayesian inference allows a machine to anchor its decisions in a mix of real observations (e.g., that you have just typed “Dear”) and informed priors (e.g., the likelihood that anyone would use Microsoft Word to write a letter). More specifically, Bayes’s theorem can give it the fluency to marshal and combine different probabilities, to reexpress, for instance, the probability that I am writing a letter if I type “Dear” in terms of the probability that I write “Dear” whenever I start a letter. Crucially, such inferences are only as reliable as the quality of the observations and the priors that anchor them, so it is vital that a Bayesian agent is intimately attuned to the current state of its world. A narrow context that provides content-level observations—whether for defusing an improvised explosive device or operating a spaceship—is better able to support timely interventions by a machine than one that is so broad that its actions are driven by superficial cues. Microsoft Office is not such a context, so let’s look at some that better fit the bill.

The witty home companion is one of three scenarios we’ll briefly look at here that unite a narrowness of focus with a wealth of contextual cues. As in the 2012 film *Robot & Frank*, AI companions will be designed to assist in the care of elderly or infirm users in their own homes, although the appeal of companionable robots has broadened significantly after the lockdowns of the COVID-19 crisis. The robot of the film alternates between carer, confidante, adviser, and friend, using modes of humor that are appropriate to each role, to reassure, cajole, and entertain as the context demands. We expect good companions to laugh at our jokes and to tell jokes of their own, but we expect honesty too. If we are to value a companion’s opinion of a good joke, it must have the wit to call out a bad one. Crucially, a good companion is an active listener: it knows when to nod and when to interject, when to laugh and when to sympathize; and when to agree or when to insist otherwise. Just as importantly, it should respect our privacy. We may confide in and gossip with our future B-9s, but they must never violate our trust. They can joke about us and to us at an appropriate time and place, but never in front of the wrong people.

This is an ambitious scenario that will be realized in small increments. Existing technologies allow machines to detect sarcasm in conversation or

on social media, and the same machine-learning techniques offer moderate coverage of irony too, as we will see in chapter 9. We may not want a robot companion to use sarcasm, but detection will be a necessity, while irony, if used for gentle mockery and self-deprecation, can foster intimacy when knowingly used by human and robot alike. Irony is easier to detect in contexts for which expectations are explicitly modeled and easier to generate when the expectations have conventional linguistic frames. As explored in chapter 7, wordplay is also easier to detect, and more appropriate when generated, if it too is tied to the vocabulary and idioms of a specific scenario.

Wordplay that pays little heed to context is gratuitous and deserves groans, not laughs. But just as a driverless car in America can learn from the near-misses of a Tesla in Japan, there are network effects to be had in computational humor too. A companion in one place that invents or acquires the portmanteau “covidiot” in the context of COVID-19 can share its acquisition with companions everywhere. So if their privacy settings permit it, our AI companions can pool their learning to remain topical and fresh. To prevent a companion from also acquiring undesirable behaviors, it must carefully monitor and filter itself, as we will see in chapter 10.

Our second scenario, a witty customer service agent, is far less ambitious in the short term. A corporate website is often our first port of call when we have a bone to pick with a vendor or a service provider, so interfacing with an artificial chatbot instead of a real human being is no longer a novel experience. We will meet the most famous chatbots in AI history, Eliza and Parry, in chapter 4 when we explore the gulf between a superficial and a deep treatment of words and intent. For now, it’s enough to appreciate that while chatbot technology has clearly advanced, so that robustly scalable and trainable statistical models now reign supreme, the guiding philosophy remains the same: a dialogue bot must still transform a user’s inputs into appropriate outputs that keep the conversation flowing, reduce any need for replies in the vein of “Does not compute” and “Computer says NO,” and satisfy the user’s needs for information and emotional support without a human in the loop.

The emotions that send us into the arms of a customer service portal are rarely positive: we come to bury Caesar’s product offering, not to praise

it. Humor can help to transform our feelings of frustration and anxiety into a more positive view of a company and its services, but only when it is relevant to the conversation and diminishes, rather than enhances, our belief that the agent’s outputs are scripted. Statistical language models, explored in chapters 6 and 7, acquire the rhythms of language for a given genre or domain, allowing machines to say the right things in the right way. If the texts on which our language model is trained also contain jokes, a model will learn to reproduce them at the right times, perhaps with minor variations. By replacing specific jokes in the training data with the generic marker <joke>, we can train a system to output this token—essentially an IOU for a joke—at suitable points in a dialogue, before replacing it with jokes that it invents itself.

Joke writing is hard, so it helps if our chosen domain is narrowly defined. As we will see in chapter 5, joke generation is a knowledge-based process that rewards a systematic mind-set, so professional comics use representations and algorithms resembling those of an AI system. Jokes are often used in dialogue to overcome an impasse—to relieve tensions, reframe a conflict, soften a criticism—and a support agent can do that if it can find a productive angle on a topic of shared concern. A witty AI agent can turn “computer says no” into “computer says no problem” by first turning “does not compute” into “does not compute *literally*,” and by looking for an angle on the topic that permits a more flexible, and more playful, treatment.

If a user complains to a hotel’s support bot, “I wouldn’t let *my dog* stay in your hotel,” then “Yes, this hotel is unsuitable for pets” is inept as a literal reply. It can, however, work as a playful reply from a machine that knows what it is doing. Key to this understanding is a grasp of sentiment or the emotional baggage of a text. As we’ll see in chapters 8 and 9, sentiment is just one aspect of meaning and intent, especially ironic intent, on which AI can use number-crunching techniques. Humor is not a quality that machines can measure directly, but if AI can quantify the many aspects that make one joke more humorous than another, it can model a numerical sense of the whole. Joking by numbers may sound chunkily mechanical, but it does allow machines to subtly weigh the merits of different joke candidates.

Sentiment plays a crucial role in our third scenario, the witty automated tutor. Virtual learning environments allow students to engage with classroom materials online, to measure their progress on tailored activities and tests, and to seek help where they struggle most. Game-like aspects can boost motivation by encouraging learners to level up with high scores, badges, or peer recognition, but gamification has its limits. It manipulates emotions yet fails to make an emotional connection when rewards are not tied to the psychological state of a learner. So who wouldn't prefer a timely joke that shows insight into a problem over a hollow merit badge, or the ability to engage playfully at the content level—with comical examples, say, that are created on demand—over the bells and whistles of a generic game? As we will see in chapter 7, in our discussion of punning, even the weakest of automated jokes can find their audience and encourage students to engage in a learning task. Chapters 3 and 4 will also show how an interactive model of causality can be used to generate playful back-and-forths and coherent stories to suit a given topic.

As in customer support, the education scenario benefits from a narrow domain focus that allows jokes to relate to topical concerns. Some may be prescribed, to be used at specific stages, but others can be generated algorithmically. IBM, which positions its Watson AI as a platform for more sophisticated bots, has explored the former with a range of joke-telling chatbots.³⁰ Here, for instance, is a prescribed bot joke that a virtual programming tutor can use to introduce its next topic: “Why did the programmer quit his job? Because he didn’t get *arrays*!” It’s a groaner, to be sure, but if used at the right time it certainly beats a cutesy level-up icon.

But it doesn’t take long for a bot to exhaust a store of prescribed gags, even if it uses them sparingly to suit the user’s context and moods. To go one better than basic gamification, a tutor must be able to generate its own jokes. The educational scenario is rather special in this respect, since the conceptual impasses that can frustrate learners have much in common with the functional inconsistencies that spur inventors to propose and patent new innovations. As we’ll show in chapter 5, problem-solving methodologies such as TRIZ encourage us to see the world as a comedian might: as a

place of contradiction and paradox waiting to be resolved.³¹ The systematic methods used to guide product innovation have obvious parallels to the theoretical concepts we will meet next, in chapter 2, and allow us to take an equally systematic view of joke creation. In this sense, the role of tutor is an ideal day job for a humorous AI: the domain provides the problems, the learners bring their conceptual impasses, and the AI generates the jokes that bridge the two.

WHEN AI COMES TO TOWN

When it comes to the public perception of AI, the dominant narrative is often one of replacement. As AI grows in sophistication and ability, machines will do more of the jobs that previously required intelligent, educated workers to perform and will do so far more cost-effectively. While the commercial reality of this narrative is hard to argue with, there is much more to the story of humorous AI than this.³²

The narrative that most AI researchers favor is one of understanding, not replacement: to really understand an aspect of human behavior, it is useful to be able to take it apart, play with the pieces, and then put it back together again. To build a machine with a human sense of humor, some disassembly is required. Engineering solutions are not always cognitively plausible or driven by the latest psychological findings, but when they yield comparable performance to a human, we are forced to rethink our explanation for how humans achieve the same ends. Many of the techniques explored in this book will surely strike you as an attempt to do an end-run around the hard problem of genuine cognitive modeling. However, as you consider where and how the techniques short-change the human mind, be prepared to rethink your assumptions about why that might be the case. If, for instance, a statistical language model or an artificial neural network can use abstruse or seemingly meaningless features to model a complex aspect of who we are, perhaps this is much closer to our own mental reality than we care to admit.

I have sketched three scenarios—a home care companion, a customer support agent, and a tutor—to give context to our discussions of theories

and techniques in the chapters to come. There is no single algorithm, resource, or technology that we can simply plug into an application to make it wittier and wiser. AI will acquire its computational sense of humor only through a complex patchwork of resources and technologies, many of which already exist in the apps on our phones. Just like those apps, the humor will be specific to the task at hand. Humor isn't a generic add-on, but a highly specific phenomenon that insinuates its way into the nooks and crannies of everything we do. Although the theories of humor we explore next must be generic to generalize, the reality is always specific. The principal insight we can take from our scenarios is that humor cannot arise in a vacuum. We must put our humorous AIs to work in serious settings that spark and inform their wit.

We will revisit the idea of a day job for our humorous AIs in chapter 10. Along the way, we explore the following topics from an AI perspective: theories of humor in chapter 2; turning theory into initial practice in chapter 3; double-acts in comedic performance in chapter 4; systematizing joke creation in chapter 5; modeling humorous incongruities in chapter 6; analyzing and generating puns in chapter 7; quantitative aspects of humor in chapter 8; and modeling sarcasm and irony in chapter 9. Once we reach chapter 10, we can look back over where our wanderings have taken us and draw some lessons about the do's and don'ts of a computational sense of humor. But for now, we start our journey at the place where all the ladders begin, the academic world of humor theorizing.

2 IT'S A JOKE, JIM, BUT NOT AS WE KNOW IT: A TOUR OF SCHOLARLY PERSPECTIVES AND THEORIES OF HUMOR

AIN'T NOBODY HERE BUT US CHICKENS

If one man's wisdom is another man's joke, then we can expect philosophers to engage in more than their fair share of witty feuds. Perhaps the bitterest and most amusing of these arose out of a clash of personalities between the ancient world's biggest stars, Plato and Diogenes.¹ Much like his peers, Plato was fascinated by the deepest questions of human existence: What is reality? What is life? What is our place in the cosmos? He tackled these questions within a system of categories and relations we call an "ontology" by placing the broadest ideas at the top and adding narrower concerns at lower levels of detail.² You might think that Plato would use humanity's most essential qualities to determine our place in his ontology, such as our capacity for language, love, rationality, or humor. Instead, he used form as his guide and placed humans under the category of bipedal animals. Since birds also shelter under this category, Plato was careful to define a human as a featherless biped. In his grand scheme, humanity would be the thing *without* feathers.

Diogenes belonged to an ancient school of philosophy known as the cynics, named for the Greek word for "dog-like." As his sobriquet might suggest, Diogenes the Cynic played the role of Rottweiler with aplomb and enjoyed few things more than snapping at the heels of stuffed-tunic philosophers. Less authoritarian than Plato and given to shocking public lapses of personal and sexual hygiene to show his contempt for prim social mores, Diogenes questioned everything.³ We humans may be featherless bipeds, but even one like himself who washed in a ditch could hope for a

nobler place in the cosmic order. Diogenes also had form when it came to talking truth to power. When Alexander the Great looked on admiringly as the philosopher did geometry in the dirt, he offered a royal boon as a token of his esteem. Diogenes merely snarled and gruffly asked the boy king to step out of his light.⁴ Humor can subvert ontologies of any kind, whether social or conceptual, and Diogenes would soon show Plato just how fragile his system of categories could be. After plucking a live chicken, Diogenes marched to Plato's academy and hurled the newly featherless biped at his rival, shouting, "Here is Plato's man."

Diogenes was a master of what Aristotle, Plato's most famous student, was to call "educated insolence." Although we can imagine him hurling his chicken with comic fury, the joke relies on emotionality taking a back seat to intelligence. Many of the labels we apply to witty people tend to emphasize their cleverness. We call them smart-asses, wiseacres, wisenheimers, wise guys, smart alecks, clever clogs, smarty-pants, and smart mouths, but implicit in these labels is the idea that one's intelligence can also be used to inflict pain on others. *Smart*, after all, can mean bright, spruce, astute, or quick-witted, but it can also denote the residual sting of a slap in the face or a cane to the palm. Diogenes set out to be smart in both senses of the word. He showed an agility of mind by jumping between the realms of the abstract and the concrete, using the latter to undercut the former and bring down Plato's ontological enterprise. Diogenes was no athlete, but this kind of mental agility requires just as much flexibility and speed and can impress just as much.

Theorizing about humor doesn't get more old school than this, and neither does AI. We can think of an ontology as a symbolic representation of conceptual norms—a shared, commonsense view of the world that we can all refer to and assume that others can refer to also—while humor makes sport of this explicit orthodoxy. In much the same way that Diogenes undermined Plato's ontology, a humorist can use devious strategies to subvert an audience's most likely expectations of a text, a situation, a routine, or a category. It is no quirk of language that AI researchers still refer to a machine's store of explicit knowledge as its ontology,⁵ or that this structure is destined to play the educated straight man to the manipulations of an insolent funny man. As we'll see in our whirlwind tour in this

chapter, humor has long been viewed as a matter of forcing a round peg into a square ontological hole. Theories differ as to what is the peg and what is the hole, or what counts as a tight fit, but all assume a rigid orthodoxy against which jokes play their insolent games. So while some of the ideas we meet here may strike you as historical curiosities, later chapters will show that each retains a modern relevance to AI and humor.

Humor theories can also differ as to how we react to a peg being forced into the wrong hole. Indeed, the question of why we laugh at all, and contort our faces into grimaces while giving up control of our bodies and senses, is one that continues to fascinate philosophers. It is certainly true that we also laugh in situations that are not humorous.⁶ We laugh when we are nervous,⁷ feel vulnerable, are relieved, or as a coping strategy in difficult circumstances.⁸ We may not be laughing *at* something⁹ in these cases, but each can still be seen as an attempt to make a serious situation more conducive to humor. As such, philosophers after Aristotle often take laughter as a starting point for their investigations of humor.

Thomas Hobbes, the seventeenth-century political philosopher, saw laughter as a mark of our sudden, if fleeting, sense of superiority over one another. In *Leviathan*, his most famous work, he writes that “sudden glory is the passion which maketh these grimaces, laughter, and is caused either by some sudden act of their own, that pleaseth them, or by the appreciation of some deformed thing in another, by comparison whereof they suddenly applaud themselves.”¹⁰ It is true that we often laugh out of a sense of superiority,¹¹ but the deformity need not be a physical one. It is just as likely to be a gross enlargement of the ego. We find it satisfying to see those who *act* superior humbled by circumstances that deliver their just deserts, and at such times we may feel rightly superior too. Yet it is an ugly generalization to view all of humor as a self-congratulatory clap on the back. This not only fails as a general theory of humor for humans; it makes even less sense for machines.

Immanuel Kant, a champion of the enlightenment, also championed absurdity over superiority, arguing that “in everything that is to excite a lively convulsive laugh there must be something absurd.”¹² For Kant, absurdity can be found in any situation “in which, therefore, the understanding can find no satisfaction,” and this Jagger-like frustration strains our faculties

as we search for its resolution. So “laughter,” for Kant, is “an affectation arising from the sudden transformation of a strained expectation into nothing.” To see laughter as an affectation rather than as a necessity is to perhaps make our physical response seem more arbitrary than it is, yet we find the seeds of two productive traditions in Kant’s take: what theorists call *incongruity*,¹³ or the degree to which we find a situation absurd, and *relief*, the degree to which we are relieved by the sudden removal of a taxing expectation.

Kant is just one link in a long chain of scholars who have identified incongruity as the vitalizing spark of comedy. The mathematician Blaise Pascal defined it as the “surprising disproportion between what one expects and what one sees,”¹⁴ while the Scottish philosopher James Beattie saw it in “things incongruous united in the same assemblage.”¹⁵ His compatriot Francis Hutcheson found the spark in a conflict of “ideas of grandeur, dignity, sanctity, perfection and ideas of meanness, baseness, profanity.”¹⁶ Expanding on Kant, the philosopher Arthur Schopenhauer offered his own ambitiously broad theory, noting with confidence that “the cause of laughter in every case is the sudden perception of the incongruity between a concept and the real objects which have been thought through it in some relation, and laughter itself is just the expression of this incongruity.”¹⁷ Take, for instance, the mismatch between Plato’s concept of humanity and the real object of Diogenes’ naked chicken. Still, while many wise heads point to incongruity as the mystery meat in a joke sandwich, none seems truly able to tell us what it’s made of. We can only hope that the demands of a computational model will force us to be more specific.

The purgative aspect of jokes, which characterizes the relief theory of humor, is also implicit in many accounts of why a purely mental clash of ideas should cause a physical eruption of emotional energy in the form of laughter. The relief theory is most succinctly summarized in a 1709 treatise by the Third Earl of Shaftesbury, *An Essay on the Freedom of Wit and Humor*.¹⁸ As Shaftesbury writes, “The natural free spirits of ingenious men, if imprisoned or controlled, will find out other ways of motion to relieve themselves in their constraint; and whether it be in burlesque, mimicry, or buffoonery, they will be glad at any rate to vent themselves, and be revenged upon their constrainers.” So can we see Diogenes and his chicken

antics as a kind of ingenious revenge against Plato and the constraints of his ontology? Perhaps. A plucked chicken has mimicry and buffoonery to recommend it, but like superiority, raging against the machine is just one more motive for humor, and this makes it less a theory of humor than a dimension along which humor occurs.

CLOCKWORK LEMONS

For Shaftesbury, the human spirit is a caged animal that just wants to be free. We are noble savages, hemmed in by society's rules and impositions, but humor lets us take back control in small but important ways. If this all sounds more poetic than scientific and leaves little room for humor in a machine that is rule-bound from the start, it chimes with a later perspective that sees humor in mechanism. For Henri Bergson, the automation of the human spirit that comes from living our lives on autopilot causes a rigidity in our dealings with others. Unlike Shaftesbury, who saw humor in the spirit's triumphant escape from its cage, Bergson finds humor in the spirit's failure to assert itself. The cage for him is not just society and its organs of control—conventions, rules, taboos, and so on—but the body itself.

In *Le Rire*, a collection of essays on comedy, Bergson discusses three qualities of the comic experience: it is strictly human, so we laugh at the nonhuman only to the extent that it reminds us of human foibles; it requires a detachment of feeling for us to laugh, so that we only feel another person's pain in miniature; and most of all, it is a social phenomenon that emerges from our relationships with others.¹⁹ Detachment allows us to strip actions and objects of their normalcy, as when, for example, we say a word over and over again to hear it as a meaningless sound. To be humorously detached is to see the ridiculous in things that society considers normal or even charming. Detachment strips lovers of their passion to turn them into grunting apes, dancers of their elegance so they become prancing clowns, and commuters of their ambitions so they become rats in a maze. For Bergson, it is the reason "we laugh every time a person gives us the impression of being a thing."

Bergson offers a Cartesian view of the individual as an agile spirit in command of an often stiff and unresponsive body. We are but ghosts

in chunky machines. In normal circumstances, we hardly notice this rift between our body and spirit, but humor emerges when they no longer operate as one. In the case of toilet humor, a loss of control over bowel or bladder can remind us of just how messy this rift can be. Or look at an old Chaplin movie: in some scenes, his plucky tramp displays an enviable virtuosity of movement, bounding here or ducking there to assure us of his graceful inner spirit. In others, he is undone by the limits of his own body, unable to duck fast enough to avoid a plank to the head or a kick in the pants.

A person who lacks imagination and always reacts to the same stimuli in the same ways can seem as rigid as any clown on a banana peel, but comic characters rarely recognize this rigidity in themselves.²⁰ Consider the characters of Niles and Frasier Crane, the snobbish brothers in the sitcom *Frasier*. Each is patronizing and conceited, yet each sees his elitist qualities as a mark of superior breeding. The show's other characters, such as Frasier's father, Martin, and his physical therapist, Daphne, are rigid in ways that complement the brothers, but none is entirely rigid. Their traits are chosen to balance a freedom in one with a risible lack in the other, and it is the subtle ways in which the show's writers tug on the ropes and pulleys of superiority, incongruity, and relief that give *Frasier* its unique comic dynamic.²¹

But this dynamic has a moral dimension too. In sharp contrast to the “no hugs and no learning” philosophy of *Seinfeld*,²² Bergson also sees humor as an occasion for personal growth. By laughing at what we find ridiculous, we encourage others to relax the constraints that suppress their nimbleness of spirit. As Bergson put it, “Laughter is the corrective force which prevents us from becoming cranks.”²³

UNHEIMLICH MANEUVERS

If laughter is the proper response to a human who acts like a machine, how are we to react to a machine that suddenly acts like a human? For Sigmund Freud, this turn can give us a rather creepy sense of the uncanny that he called the *unheimlich*.²⁴

The *uncanny* or the *unheimlich* hinges on a category error. Diogenes' chicken is a category error, since a plucked bird seems to fit into two disjoint

holes at once: the category of things that are human and the category of things that are not. It is natural to link the absurd with comedy and the uncanny with horror, but we can also view the latter as a joke that prickles rather than tickles. Just think of the most enduring horror tropes: intangible ghosts that wreak havoc in the physical world; zombies and vampires that are dead *and* alive; werewolves that are both man *and* beast; dolls and ventriloquist dummies that speak for themselves; and, of course, robots that walk and talk like us but are clearly not us. The horror genre abounds in category errors, and Freud loved it. He was especially fond of the peculiar air that pervades the tales of E. T. A. Hoffman, in which eye-slurping demons, creepy doppelgangers, and human-like dolls discomfit the reader.²⁵ As a proponent of relief theory, Freud saw the value of psychological release when challenges to our systems of pegs and holes are framed as harmless make-believe.²⁶

Category errors are always incongruous, whether they are comic or uncanny, so the unheimlich is just a small push away from becoming laughably ridiculous. Mel Brooks and Gene Wilder brought out the silliness inherent in Mary Shelley's *Frankenstein* while being meticulous in their recreation of the Universal Studios movies of the 1930s. Their younger Frankenstein has zippers instead of sutures to allow for easy after-market modifications, and Wilder's mad scientist teaches his creation to sing and dance in a top hat and tails. Roman Polanski reinvented the vampire movie aesthetic in *The Fearless Vampire Killers* and gave us two new subspecies of bloodsucker: a Jewish vampire who scoffs at Christian crucifixes ("Oy! Have you got the wrong vampire!") and a gay vampire who lusts after Polanski rather than his female costar. The category errors of the horror genre are not funny in themselves when they are treated nonsatirically by a film that takes itself seriously, perhaps because well-made films take care—in Kant's terms—to not overstrain our expectations. But since the incongruities are already halfway to being funny, it is not surprising that we continue to draw on them for comedy.

Whenever we stitch eclectic chunks of knowledge into a composite whole, we build our own conceptual monsters and become our own Victor Frankensteins. Category errors like these are a special case of what theorists call a conceptual blend.²⁷ As formalized in conceptual blending theory (CBT) by Mark Turner and Gilles Fauconnier, a blend uses various

constraints and principles²⁸ to coherently combine elements from multiple inputs, called *mental spaces*.²⁹ CBT has proven especially useful in the analysis of creative language, from metaphors and poems to advertisements and jokes, but the latter rarely give audiences a polished integration of frames. In fact, rather than taking pains to hide the joins, jokes draw our attention to them.³⁰ They saddle us with a sudden need to alleviate a strained expectation, yet also give us the sense of ridiculousness, superiority, or relief that arrives with its resolution. As Seana Coulson, a cognitive scientist and CBT theorist, has argued, the punch line of a joke alerts listeners that the frame on which they have pegged their understanding is unequal to the task, and so it is time for a radical switch-up. This frame shifting is a game-changer that happens late in the game, and brings with it a fleeting panic that is soon resolved by an apt choice of alternate framing.³¹

Consider a joke from comedian Emo Philips: “I love to go down to the park and watch the little kids skip and jump. They don’t know I’m using blanks.” Metaphors and jokes both ask us to attend to the overlaps between two realms of experience, but metaphors use this overlap to foster a greater appreciation of a topic, while jokes use it to briefly deceive us. The overlap in the frames Playground and School Shooting is neither deep nor especially edifying, yet it is enough to force us into an emotional U-turn. As we shift frames, we carry baggage from one into the other. What results is a counterfactual scenario with aspects of both and the truth of neither. Consider this exchange in the film *Jurassic Park*, in which the scientist Ian Malcolm chides the park’s owner, John Hammond, for the omnishambles that it has inevitably become.³² As the park’s test-tube dinosaurs run hungrily amok, the survivors huddle in a cafeteria with only melting ice cream for comfort:

John Hammond: All major theme parks have delays. When they opened Disneyland in 1956, nothing worked!

Dr. Ian Malcolm: Yeah, but, John, if *The Pirates of the Caribbean* breaks down, the pirates don’t eat the tourists.

Malcolm’s blend is a metaphor (*Jurassic Park* is Disneyland) that sours into a joke. He wants us to see the joins that Hammond is so desperate to cover up. Blends are more general than jokes or metaphors, yet we might still

expect a computational model of blending to serve as a useful precursor to an AI model of joke creation.³³

Like incongruity, an idea as versatile as blending does not arise in a single place at a single time, but pervades the work of scholars in different fields and eras. One especially influential forerunner to CBT, *bisociation*, was proposed by the intellectual Arthur Koestler in the 1960s to explain not just the creativity of art and science but of comedy also.³⁴ The mental association of ideas leads us from one thought to another in an intuitive if often automatic fashion, as though we were pulling on a thread, so we jump from hopping and skipping to children's games, or from bullets to guns and from guns to shooting. But Koestler argued that the basis of creativity is not one-way association but two-way bisociation, that is, the ability to situate an idea or an experience in two frames at once. For Koestler, bisociation occurs in the narrow overlap between two frames of thought—which he quaintly named *matrices*—that are commonly imagined to have no overlaps at all.

Bisociation also contains shades of the relief theory favored by Shaftesbury and Freud. As a moment of insight emerges from the electrical signals of the brain, the body often lags behind, especially when the insight provokes a rapid change of emotional perspective. The chemical signaling of the body is simply no match for the neural switching of the brain, and so we gasp at jump scares in horror films, or at jokes that send us this way and that, because the body needs to shrug off the physical tension that the mind has decided is no longer necessary. This goes some way to explaining the overlap between the “ha-ha” of comic insight and the “Aha!” of scientific discovery.³⁵ Famously, the bathing Archimedes had so much nervous energy to dissipate when his eureka moment came that he ran naked through the streets of Syracuse. If machines are not so physically invested in the fruits of their mental labors, can they ever truly appreciate a joke in the same way as us?

Other theories of conceptual overlap can be situated between bisociation and CBT in the family tree of humor theories. Victor Raskin's semantic script theory of humor, the SSTH, was first outlined in his 1985 book, *Semantic Mechanisms of Humor*.³⁶ The SSTH speaks not of frames, spaces, or matrices, but of scripts. When we visit the doctor, order lunch, or get mugged in a dark alley, the action follows a certain, almost mechanical

order.³⁷ Things work they way they do for a reason; we feel ill *before* we visit a doctor, feel hungry *before* we visit a restaurant, look at the menu *before* we order food, and eat our food *after* it has been brought to our table. Bergson might well spy a rigid automation in the actions of our daily lives because convention and experience have fixed the order that makes the most sense. After all, we have good reason for putting on our socks before our shoes.

For Raskin, a text is not objectively humorous in itself because tastes differ as to what is laughable. Instead, a text—whether spoken, written, or performed—has humorous potential if it is compatible in part with a pair of conflicting scripts, with the dominant one foregrounded while the other remains a dormant possibility. Jokes gull us into activating the dominant script to explain an event, then force us to switch to the alternative when an incongruity makes it difficult to do otherwise. If script-switching sounds a lot like frame-shifting, that's because both model the same process as viewed from different perspectives. Script-switching focuses on the time course of joke interpretation and the awkward reorientation that is needed to overcome its surprising impasses. Frame-shifting works similarly, but emphasizes the blended perspective that a joke requires us to adopt.

Consider this one-liner from Will Marsh, which was ranked among the top ten jokes of the 2012 Edinburgh Festival Fringe: “I was raised as an only child, which really annoyed my sister.” Even one-liners can fit the two-script mold of the SSTH:

Script 1: Being raised *as* the only child in a family with just one kid
 (“as” = “is”)

Script 2: Being raised *like* the only child in a family of several kids
 (“as” = “like”)

Humorous script changes often hinge on seemingly insignificant aspects of a text. Marsh’s quip exploits two competing meanings of the word *as*, and our default preference for one (“as” = “is”) over the other (“as” = “like”) when we don’t smell a metaphor. Yet what makes the audience laugh is not the shifting sense of the word *as*, but the shifting emotional dynamic as we switch from script 1 to script 2. Script 1 codifies our sense of what it is

to grow up without siblings, while script 2 codifies our sense of what it is to grow up with other siblings in our shadow.

Marsh's joke taps into a wealth of tacit knowledge about what it means to raise a family. If we were to use a conspiracy theorist's pins and colored yarn to map out the connectedness of this knowledge, we would build a network much like the one shown in figure 2.1. Few of these ideas are mentioned in the joke itself, so we must unearth them for ourselves. It is the job of a humor theory to give formal shape to the results of our introspections, but it is the job of a computational theory to turn those abstract shapes into explicit data structures—with specific storage and retrieval protocols—that machines can use to put the formal theory into practice.

Raskin was careful to note that a humorous switch will hinge on scripts that differ in a crucial respect he named the *script opposition* (SO).³⁸ Since a great many scripts overlap and differ in unfunny ways, Raskin argued that only certain SOs give a text the potential for humor. Just what the most

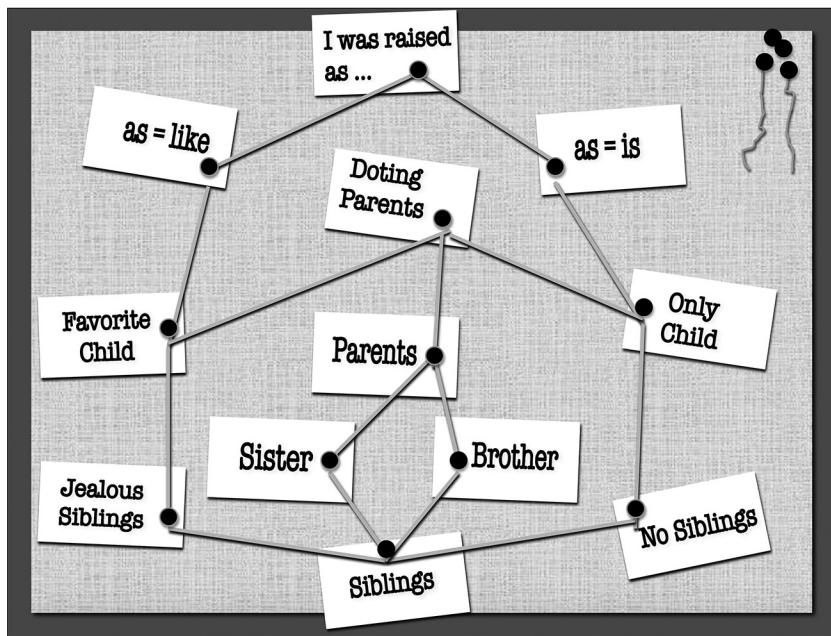


Figure 2.1

Semantic connections between the ideas that are implicit in Will Marsh's joke.

effective SOs might be is a matter for empirical inquiry. If we look at the jokes that make us laugh and pinpoint the SOs that make them work, we find distinctions like pride versus shame, life versus death, wealth versus poverty, sex versus innocence, and even some versus none (as in siblings) again and again. When Malcolm compared Hammond's ill-fated Jurassic Park to a Disneyland ride that eats tourists, he tapped into a range of SOs with a proven track record in jokes, from life versus death and safety versus danger to pleasure versus pain and order versus chaos, not to mention good versus bad, smart versus stupid, and clever versus unwise.

Working with Salvatore Attardo, Raskin later expanded his script theory into the *general theory of verbal humor* (GTVH).³⁹ If the STH is a kitchen appliance with just a single SO attachment, the GTVH gives it other cool attachments too, such as SI (situation), LA (language), NS (narrative strategy), TA (target), and LM (logical mechanism). Each allows a different knowledge source to be plugged into a joke, but it is the LM that contrives to bring together two scripts in a single text. For instance, an LM named *figure-ground reversal* is responsible for the deliberate misdirection in the old joke about a factory worker who appears to be smuggling goods past the guards in a wheelbarrow, but is really just stealing wheelbarrows. More than any other module, the LM gives a joke its distinctive character and stirs a sense of *déjà vu* for others. When we say, "Stop me if you've heard this one," we really mean, "Tell me if you've heard a joke that uses the same LM for a similar effect."

Another tune-up by Raskin, Attardo, and their colleagues installed a much more flexible notion of script into the GTVH. Out went the rigid this-before-that model of human affairs that sought to capture the Bergsonian automation of our lives, and in came the idea of scripts as generalized graph structures that look more like our pins-and-yarn analysis of Marsh's *only child* joke.⁴⁰ Computer scientists use graphs to capture the connectedness of ideas, so rethinking scripts as graphs allows the GTVH to build on a wealth of AI research that takes a graph-theoretic view of, for example, analogical and metaphorical reasoning. It also reconciles the GTVH to Koestler's quaint notion of bisociative matrices. When a graph is stored as an adjacency matrix—a table of rows and columns in which two nodes, A

and B, are connected if the intersection of row A and column B contains a nonzero value—script overlap and matrix bisociation begin to look more like kindred notions.

More recent revisions of the theory build on foundations that reach all the way back to Plato. Raskin describes the *ontological semantic theory of humor* (OSTH) as a theoretical grandchild of his original semantic script theory.⁴¹ While the GTVH's modularity was driven in part by a desire for interdisciplinarity, so that scholars of different stripes might contribute to its various attachments, the Osth affirms the primacy of linguistic semantics to the workings of jokes. All aspects of the general verbal theory, from scripts to LMs, can now be folded into an all-embracing model of text interpretation with a well-engineered ontology of words and their meanings at its core. Think back to our pins-and-yarn graphing of the ideas in Marsh's only child joke. It is the job of a semantic theory like the Osth to specify how those ideas can be accommodated in explicit, frame-like structures and to provide a procedural means of mapping from words into these structures.

Raskin is bearish about the wholesale adoption of statistical data crunching as a substitute for symbolic structures like these, and refers to the displacement of old school semantics as AI's "statistical winter."⁴² Nonetheless, the real value of an ontology ultimately resides in its ability to unify disparate perspectives around a common representation of meaning. As such, the statistical models we explore in this book can still work within a broad approach to humor that includes the Osth.

B-9 VIOLATIONS

Every theorist brings a particular focus to humor. Like rival archaeologists digging in the desert, each may unearth fragments of a different beast while the sphinx that unites them all remains buried in the sand.⁴³ Take, for instance, the idea of incongruity. If a state of affairs violates a moral principle, or even the principle of cause and effect, then we might consider it incongruous. But "violation" implies the infringement of a governing code, and a theory based on violation rather than incongruity can also add a top note of disapproval to this underlying shock value.

According to the *N+V* theory of Thomas Veatch, we often laugh at a violation (V) that is also, paradoxically, quite normal (N).⁴⁴ In this view, N+V jokes embody Hamlet's philosophy that "there is nothing either good or bad but thinking makes it so." In line with Aristotle's original claim that we tend to laugh at "some defect or ugliness which is not painful or destructive,"⁴⁵ there is a degree of relief, and a hint of relief theory too, in seeing that a violation is not so harmful after all. In a similar vein, Peter McGraw and Caleb Warren's theory of *benign violation* views *normal* as just a special case of *benign*, which, following Aristotle, is anything that does not bring us pain or destruction.⁴⁶ Indeed, the greater the apparent violation and the more benign it turns out to be, then the greater our readiness to laugh at this revolution in miniature. Although this is a rather high-concept idea, it makes some testable predictions, and McGraw and his team at the Humor Research Lab (HuRL) at the University of Colorado have conducted a number of psychological experiments to show how our perceptions of humor tend to vary in proportion to our perception of how transgressive *and* how harmless it all seems.

But relief is just one possible reaction to a violation made suddenly normal or benign. Learning is another. A logical view of the world gives us concepts with sharply drawn boundaries that jokes gradually smudge into blurred lines. Jokes reveal to us the boundary cases in our reasoning—the special circumstances where the rule breaks down in favor of its exception. Just as new data force theorists to adapt and ruggedize their favored frameworks, jokes push us to revise the mental representations that led us to places of incongruity or violation in the first place. If humor theories can evolve, why shouldn't our scripts grow with experience too? Bergson saw laughter as a cue to realign the body and spirit, and his chiropractic view finds its technical equivalent in a theory advanced by one of AI's founding fathers, Marvin Minsky.⁴⁷ When we laugh at the stupidity of others, we diagnose the causes of this stupidity by sensing the limits of our own common sense. So, for Minsky, every joke is a call to pull out our mental lug wrenches and get to work, to tighten here and loosen there so that our representations do not also lead us astray.

The relief dimension of humor is, in a sense, the pleasure principle of humor. It is the mind's reward to the body and itself for making lemonade

from someone else's clockwork lemons. As Matthew Hurley, Dan Dennett, and Reginald Adams have argued, humor may be an evolutionary adaptation that rewards us with mirth (the joyful feeling imparted by humor) for revising or undoing our faulty beliefs and inconsistent models of the world.⁴⁸ In a sense, jokes are the humor equivalent of pornography, since it is human nature to try to game any system that gives us pleasure. For Hurley, Dennett, and Adams, jokes are "supernormal stimuli," that is, skillfully heightened instances of the kind of stimuli we find in our normal lives, just as pornography concentrates those qualities we find sexually alluring.⁴⁹ So we trade jokes the way kids trade racy magazines, while an objective definition of humor remains just as elusive as a hard-and-fast rule for pornography.⁵⁰

By now you will have noticed something of a trend in how humor theories, especially those that hinge on absurdity and surprise, are named. Take the theory with the broadest tent and consider its full name: *incongruity resolution*.⁵¹ What is really so incongruous about a comic situation if the incongruity turns out to be so amenable to resolution? The trick resides in the ordering of the words: we first encounter an incongruity, *then* find its resolution, so that the humor emerges in a two-stage shift from panic to revelation.⁵² The folklorist Elliott Oring prefers the label *appropriate incongruity*, since comics have a knack for explaining why that round peg really does belong in the square hole.⁵³ Salvatore Attardo, a cocreator of the general verbal theory of humor, opts for the label *relevant inappropriateness* when theorizing about irony.⁵⁴ An ironic remark seems inappropriate in the context in which it is made, but becomes relevant when we peg it to a context in which it would be more apt. Thomas Veatch favors N+V, a violation of norms that reveals the normalization of violation, while McGraw and Warren opt for *benign violation*, an apparent shock to the status quo that turns out to be not so shocking after all.

Each pairing is an enigmatic, two-word oxymoron that acts as a calling card for the theory it names. Like "kosher pork" or "cold fusion," each hints at something silly and irreconcilable in its shotgun marriage of opposing ideas. For his part, George Orwell reduced humor to the pithy "dignity on a tin-tack."⁵⁵ The ancients would have seen this as a kind of *sympathetic magic*—the idea that artifacts, like effigies or fetishes, have power to affect

us when they imitate that which they seek to master.⁵⁶ What better way to identify a theory of humor than with a name that captures something of the ineffable magic of jokes that it sets out to explicate?

BIT PLAYERS

As programmers, we can dictate the terms of laughter to a machine, to stipulate that it will laugh at this or that arrangement of stimuli. The laughter that ensues won't be organic laughter, but the false laughter that talk show audiences are prompted to produce with cue cards and flashing lights. Still, false laughter is all part of being a socialized human.⁵⁷ We all do it, whether it is chuckling politely at the jokes of others or smiling our assent to the views of the group. This isn't the kind of laughter that hijacks our bodies or swells into a physical need to slap our thighs and clutch our ribs. This is the controlled laughter of social signaling, and it is as much an artifice as the print ("ha-ha") of a computer program or the hashtag #HAHA of a tweet. There is no doubt we can train a machine to capture this deliberate kind of laughter in its social interactions with humans. But to return to a question I first posed in the context of bisociation, can we ever give our machines enough skin in the game to truly feel tickled themselves?⁵⁸

If so, we will need to give machines two modes of appreciation for two forms of laughter: the controlled and carefully emitted variety versus the uncontrolled and genuinely evoked variety. Neuroscientists denote the latter, which arises as an emotion-laden response to a stimulus, as Duchenne laughter; this is laughter that can be read on the face, in the characteristic muscle movements around the mouth and eyes. In contrast, non-Duchenne laughter is a voluntary act; since it is ungrounded in spontaneous feeling, it omits the poker tells of the real deal.⁵⁹ Unsurprisingly, each kind of laughter is processed in different ways via different neural pathways.⁶⁰ So just as we can instinctively tell a posed smile from an authentic one, we tap into different intuitions about laughter and its effects on others to discern when someone is feigning mirth or genuinely experiencing it.

It's not just comedians who have an incentive to tell one from the another. We all do. We each benefit from the masks we wear in public and

from an ability to peer behind the mask when it really matters. Authentic laughter reveals itself. It is embodied and unbidden. It relaxes our muscles, interferes with our breathing, and induces fits that can leave us feeling physically helpless.⁶¹ The evolutionary benefit conferred by this loss of bodily control is far from obvious⁶²—it may be a means of promoting the play that is key to our cognitive development or of hindering activities with serious outcomes⁶³—but it does suggest that authentic laughter is a deep-seated, organic part of who we are.⁶⁴ If controlled laughter is just the polite icing on the cake, uncontrolled laughter is baked in from the start.

But baked into what? A fascinating possibility is offered by a general theory of insight from the AI researcher Jürgen Schmidhuber, which goes to the core of our success as cognitive agents.⁶⁵ Our ultimate goal is to understand the world in which we live so that we can survive and thrive within it. This understanding requires an ability to explain the past so that we can categorize the present and predict the future. By seeing the general within the specific, we can discern the hidden patterns that connect seemingly disparate objects or events and exploit those patterns to go from cause to effect and from insight to action. A cognitive agent is motivated by its own survival, and this is bound up with its ability to reliably distill raw information to its essence. The measure of how well we grasp a situation is how much predictable detail we can strip away to arrive at this essence, so in computational terms, Schmidhuber proposes a data compression view of insight. As illustrated in figure 2.2, he paints a picture of understanding as an ability to squeeze predictable meaning from each new stimulus that we encounter.

Data compression gives us an information-theoretic basis for measuring how much insight an agent shows in any given situation. To see how, imagine how a computer might save an image of a checkerboard to disk. A naive program that treats every pixel as though it could contain any color allocates twenty-four bits to each pixel, while a less naive program, noticing that the image uses just two colors, black and white, allocates a single bit to each pixel. The most insightful program, however, recognizes the image as a checkerboard and simply stores the color, size, and upper-left coordinates of each square. In pure bit-counting terms, the second program is twenty-four

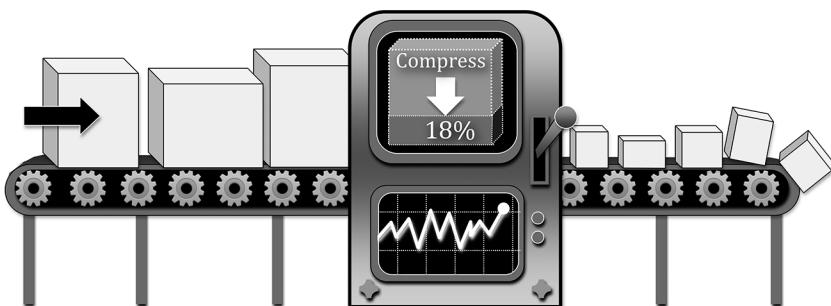


Figure 2.2

Our ability to extract meaning from new stimuli is given a quantifiable form in a data compression view of cognition. Predictable forms are the most compressible.

times more insightful than the first, since it achieves a twenty-four-fold compression of the image. But the third program is vastly more insightful still: for a large enough image, it can achieve a 1,000-fold compression.

Insight permits generalization, and generalizations allow us to squeeze new stimuli into familiar patterns, so we can quantify an insight using the number of bits saved by compression. Schmidhuber imagines that cognitive agents use adaptive methods to find recurring patterns in the stimuli they are exposed to.⁶⁶ Let's suppose that the stimuli are faces, and the recognizer generalizes well over a diverse range of human faces that an agent is likely to see in a typical day. If, as shown in figure 2.3, we now expose the agent to a succession of novel stimuli that correspond to comical or non-human faces, compression rates will drop sharply until the recognizer can adapt to the new normal and learn to generalize over the recurring features of the new data. Schmidhuber views the compressibility of a stimulus as its subjective momentary simplicity, as this sense of simplicity will shift over time as we learn to see the familiar in the surprising.

A good joke is like a wax apple to a fruit fly, or a misleading use of a recurring pattern to a compression algorithm. Jokes confuse the compressor by disguising an instance of one pattern—a script or a frame—as an instance of another. Prior to grasping its essence, poor compression is achieved on a humorous stimulus when the number of caveats to the misidentified pattern outweighs the savings gained from its detection. However, once the

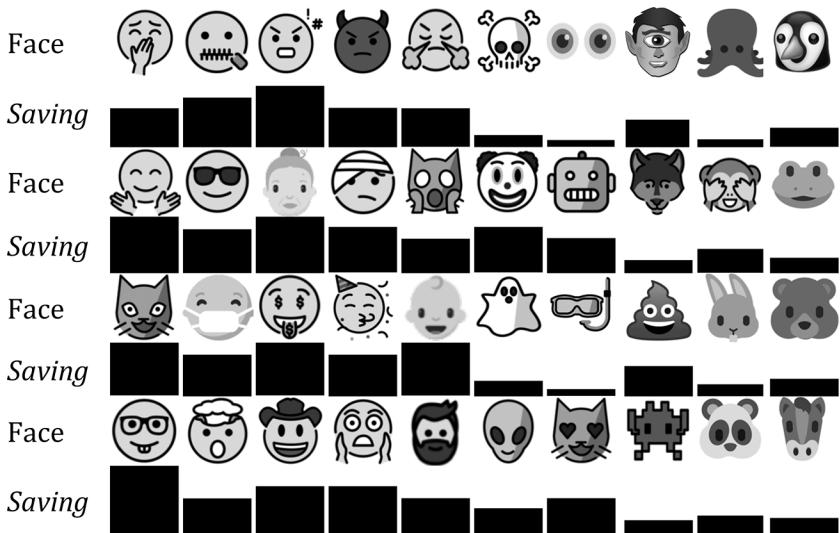


Figure 2.3

Saving Face: A face detector recognizes and compresses a sequence of new stimuli. The black bars indicate the compression achieved and the number of bits saved.

true pattern is discerned, the agent can better compress the stimulus and extract a quantifiable value. Schmidhuber applies a variety of labels to this time-dependent difference, from “novelty” and “surprise” to “interestingness,” “aesthetic reward,” “internal joy,” and even “fun.”

So, it is only a short leap from “Aha!” to “ha-ha.” As the initial incongruity of a punch line melts away—or, in Kant’s words, there is a “sudden transformation of a strained expectation into nothing”—the extent of the incongruity, and of the relief that its resolution brings, can be measured in the number of bits that the compressor has managed to save by actually getting the joke. It is on this saving that authentic laughter can be evoked within the agent. The greater the saving, the bigger the laugh. This is hardly the joy unconfined that most humans would recognize, no matter how many bits are saved, but large savings do accord with the “Aha” sensation that can accompany creative insight.⁶⁷ There is undoubtedly more to beauty, joy, humor, or fun than a sudden realization of compressibility, but this is nonetheless something real, and quantifiable, from which a machine can derive

a truly useful sense of intrinsic motivation. While a few shaved bits might seem a weak cause for celebration, much less a roar of approval or a rush of endorphins, it's a start. Toeholds like these sometimes become beachheads.

BASELINE VOLLEYS

Humor theorists tend to agree on very little across the board, but at least their competing theories give us different nouns for different clowns. As we have seen, the three most loaded nouns are *superiority*, *incongruity*, and *relief*.⁶⁸ Clowns of the first kind make sport of power structures, in ways that enforce or subvert the status quo. Clowns of the second kind derive humor from opposition, whether of sense and nonsense or norms and their violations. Clowns of the third kind treat humor as an escape valve for the pressures of life, and so they puncture pieties, tweak taboos, and generally turn lemons into lemonade. It is tempting to imagine that each kind performs in a separate part of the three-ring circus that is humor, but jokes that exhibit just one kind of humor are rare indeed. Just as incongruity jokes typically raise tensions before relieving them, superiority jokes can ascribe absurd mind-sets to their targets or use incongruity to invert social hierarchies.⁶⁹

What are the baseline requirements for giving each of these aspects of humor to a machine? For a machine to be in total command of any particular one, such as incongruity, it must be able to predict the ramifications of blending one idea with another, or of framing one as another, at least as far as an audience is concerned. Yet total command is not always possible, even for an expert, and we can settle for less. If we position our humor generator in a sweet spot that trades some control for serendipity, to foster rather than command aspects like incongruity and relief, then we can model them in miniature with relatively simple generation processes.

In the next chapter, we'll see how Twitter bots—automated users of the Twitter platform—exploit some of the simplest ways of turning humor theory into humor practice. Like any human user, a bot can read the tweets of others or post tweets of its own. Relatively few bots pretend to be human, and fewer still could sustain this pretense for very long. Rather, most users knowingly follow a Twitterbot for the “otherness” of its voice or the

peculiarity of its algorithmic fixations. The bots we meet next favor unpredictability over reliability and wring humor from the simplest methods and resources. Twitterbots can be as basic or as ambitious as we care to build them and allow us to reuse disparate resources in thought-provoking ways.⁷⁰ In addition to presenting bots as a baseline for an AI treatment of humor, we will also get stuck in and build some of our own by using some tools that make the experience of building humorous bots fun, easy, and rewarding.

3 TWEET MY SHORTS: TWITTERBOTS CAN TURN OUR THEORIES INTO SIMPLE PRACTICE

CHILD'S PLAY

Scholarly theories paint a grand vision of humor, but it still falls to AI developers to execute this vision in code. If we are going to start small, we will need to think small too. In this regard, it can pay to think like a child. Kids don't overcomplicate humor, and they naturally gravitate to the simplest means of making a joke.

A key challenge of computational modeling is giving machines the knowledge to understand and make jokes. The more conceptual the humor, the more likely it is to hinge on our shared knowledge of scripts and frames, but a child's humor is rarely so profound. Although it is difficult to avoid these structures in language, since they give shape to the meanings that lurk between its surfaces, it is at the surface that children tend to play. Puns, for instance, rely on simple substitutions of phonetically similar words, even if the best nudge us from one frame to another as we consider the effects of a replacement. But we don't need to appreciate all of the ramifications of a change to laugh at it, and some puns even come with their own training wheels, in the form of a helpful template. Consider the knock-knock joke, in which one of the joke's two scripts is built into its formulaic structure:

S: Knock Knock

L: Who's there?

S: Kanye.

L: Kanye Who?

S: Kanye let me in already, it's freezing out here.

The knock-knock's training wheels can support children *and* machines, making it an ideal playpen for training a computer to generate jokes of its own.¹ It doesn't take much of an education to be playfully insolent with the formula; the flimsiness of the switch-up ("Kanye" as *Kanye West* versus *Can you*) even adds to the insolence of the result when it drags a famous name into the joke too. Perhaps it is just as well, but nothing we know about anyone named Kanye ever makes it into the joke.

Nonetheless, we can trade in the pun for some brand recognition, to poke some fun at a well-known individual and at the knock-knock formula itself, as in:

L: Kanye Who?

S: *Kanye West. How many people named Kanye do you know?*

The trailing "do you know?" is especially salient here. Jokes allow tellers to show off what they know and to remind audiences that they know it too. So jokes don't so much rely on a knowledge gap as on a salience gap, since tellers and audiences have varying expectations as to how this shared knowledge is relevant in context. It may not seem like much, but the variant above hinges on two pieces of content: that *Kanye* is a rather rare name and that it identifies a famous individual. If we stick to a pun-free formulation of the knock-knock joke and use the punch line to shoehorn in some specific content about a well-known figure, we can introduce a second frame *and* a second script into the joke, as in the following variant:

L: Kanye Who?

S: *Kanye West. Can you move your car? It's blocking my Lamborghini Urus.*

The frame in this case is that of *Kanye West*, which is a store of interesting tidbits about the man's public persona and material possessions. The script, in which one parked car is blocked in by another, overlaps nicely with the knock-knock formula. This may not spark an incongruity, but it does offer a potential for conflict. There are plenty more where this came from. Just imagine what drama a stranger might bring to your door with a sudden knock and then turn it into a script as follows:

S: *Kanye West. Tell Kim that I've signed the divorce papers.*

S: *Kanye West. I've brought the ransom money. Now give me Kim!*

S: *Kanye West. Outta my way before I hurt you with my massive ego.*

S: *Kanye West, of course. Didn't you recognize my leather suit?*

S: *Kanye West. Adam West is busy "playing Batman" with your momma.*

And so on. Each script is an opportunity to showcase a different part of what we know about the key figure in the joke. This second script is deployed on schedule, right after the “who” of the primary script, but we can also preempt its arrival:

S: *Kanye.*

L: *Kanye West? Please go away. We agreed that this is Taylor Swift's night.*

Let's just unpack what we have done here, using the terminology of the theories in the previous chapter. Answering a knock on the door is, ostensibly, the fixed primary script of a knock-knock joke, while the doorstep dramas that ensue once the door is opened are its sources of variation, its secondary scripts. Since the first script is a contrivance to activate the second, the joke turns on the bisociation of that other script with the frame that instantiates it. Here we see the real difference between a script and a frame. The former is a generic sequence of events, while the latter is a set of specific values that can flesh out the roles and rationales of those events. Several script representations were discussed in chapter 2, from sequences of events to generalized graph structures, but when our aim is to realize a script as a piece of text, simple templates can work just as well—for example,

```
"#frame.name#. Tell #frame.spouse.firstname# I've brought the divorce  
papers."
```

Templates map scripts directly into language, leaving slots for our frames to fill. If we encode the various doorstep dramas of a knock-knock joke as templates, then we can fill them with our pick of frames. We'll shortly meet a resource called the NOC that gives us an abundance of character frames, each with a bundle of facets to fill our templates. If we define T templates,

and if the NOC has C characters, and if the average number of usable fillers per facet of each (e.g., different clothes for Kanye, different cars for Kim) is B for branching factor, then we can generate TBC variations of this character-driven, nonpunning strain of the knock-knock joke.

While TBC may be large, we do not have to hear all TBC variations to tire of the formula. Templates and frames define a closed world that is fun at first, but its walls soon close in. Still, we can trade the control offered by closed structures like these against the freedom of unknowable open-world events. By giving up most of our knowledge and much of our control, we can embrace a more superficial model of humor generation in which machines produce fresher results with less. These surface-oriented approaches, as exemplified by the Twitterbots we will meet next, take us away from the conceptual levers of scripts and frames, but we can still aim to foster the incongruity, superiority, and relief that are characteristic of jokes. As for the frame-driven knock-knock jokes we have discussed here, we'll revisit them at the end of this chapter when we will have the tools to make them a reality and the opportunity to compare the open- and closed-world approaches to humor.

GUILTY PLEASURES

If knowledge is power, then a small gap in knowledge is the minimal information-theoretic basis for one agent to feel superior to another. An informed system can assert its superiority by hurling tiny nuggets of insight at those who seem to lack it. As shown by @stealthmountain, a now-suspended Twitterbot, no insight is too small to convey an air of superiority when it also levels a criticism at its target.² To prove the point, this bot used nothing more than the spelling of “sneak peek” to target users who mistype “peek” as “peak.” It’s an easy mistake to make, so the bot gently chided those who did with this passive-aggressive reply: “I think you mean ‘sneak peek.’” Since politeness is just one of the ways we put a benign veneer on acts of social superiority, few ever showed gratitude for interventions like these. Indeed, many reacted with furious indignation. Nobody likes a busybody, especially one who is right, and within this tiny sliver of linguistic

performance, the bot really was superior to its targets. Viewing the bot as a human provocateur—its name gives little away³—some enraged targets labeled it “a spelling fascist” or hurled at it a level of abuse that would seem excessive if directed at an online predator.

This is where the superiority humor of @stealthmountain really shows itself: not in the bot’s own actions but in the reactions of its targets. In Bergson’s terms, the bot is as rigid a mechanism as it can be. It always responds to exactly the same stimulus in exactly the same way, since it is deaf to the context in which it finds its trigger phrase. While some targets react with gentle humor rather than peevish scorn, as when a prominent comedian replies, “I am on back medication. It looked right to me,” many more are humorous despite themselves.⁴ Indeed, their hair-trigger reactions reveal the bot’s human targets to be almost as rigid as the bot itself. It’s precisely because the stakes are so small that these overreactions strike us as funny. When one target talks of dissolving @stealthmountain in acid, if only to show how little impact its loss would have on the world, our sympathy for his irritation does not stop us from laughing at the verbal extremes to which this user has been moved.

Bots like @stealthmountain are the mosquitoes of the digital world: as small a nuisance as they actually pose, some still react as though their bites carry malaria. @stealthmountain comes off as superior to most of its targets, yet it’s a benign bot that does no more than signal a spelling error. Other bots wield their information-theoretic superiority in less benevolent ways. Consider @enjoyTheMovie, another defunct bot that leverages a store of film spoilers to lord it over its chosen targets. A user need only mention a film from its database in an aspirational context for the bot to respond with a demoralizing spoiler, such as that Darth is Luke’s father or that Bruce Willis was dead all along. Although few ever thanked the bot for its unsolicited insights, the shrieks of anguish it earned from its targets surely heightened the vicarious thrill for its followers. Those who actively followed the bot may have been indifferent to its spoilers, but they could still feel superior to those whose cinematic knowledge fell short of their own. Superiority humor is supposed to be divisive, and it often plays to that part of an audience that is flattered by the joke.⁵

It's no coincidence that neither of these bot provocateurs is currently active on Twitter, since bots that make unwanted interventions into the timelines of others quickly fall foul of Twitter's Report Abuse button. In contrast, bots that set out to create occasions of comical incongruity rarely impose themselves on others. Instead, the most popular incongruity bots generate collisions of ideas that bring the rubberneckers *to them*. Take, for instance, Darius Kazemi's @twoheadlines, a bot that uses topical headlines as vehicles for its linguistic car crashes.⁶ Like many bots that aim to produce novel texts of a familiar form or genre, @twoheadlines uses the cut-up method made famous by Brion Gysin and William S. Burroughs.⁷

Gysin generated his first cut-ups by accident while using a stack of newsprint as a cutting board for his collage work. Intrigued by the random juxtapositions he observed in the jumble of scraps, he and Burroughs eagerly took a knife to texts of all kinds. After Burroughs cut up a text about the tycoon J. P. Getty to generate, "It's a bad thing to sue your own father," he was moved, a year later—once Getty was in fact sued by his own son—to suggest that "when you cut into the present, the future leaks out." Kazemi's operating principle in @twoheadlines is more practical than oracular: when you cut into the news, incongruity often leaks out.

True to its name, his bot works with tweet-sized headlines that can be sourced in abundance on the Web, from services like Google News, and cuts up a random pairing of any two to produce an imaginary third. But unlike Burroughs and Gysin, Kazemi's bot is more surgical in how it places its knife, opting to cut headlines at the boundaries of named entities, such as "Elon Musk" or "Space X." Followers of @twoheadlines never actually see the two headlines that go into each cut-up. They see only the end result, in which a named entity in one is replaced with a named entity from the other. We can only guess at the original filler and at the discarded headline that supplied it. The goal is to swap out a contextually appropriate entity for one that is both contextually inappropriate and appropriately incongruous in its new setting. Since the bot has no semantic model to guide it in picking the most comical substitution, it simply plays a percentages game: it tweets enough cut-ups for some of them to be humorous, and

then relies on its followers to retweet the wheat from the chaff. The following tweet is perhaps my favorite cut-up from @twoheadlines:

Miss Universe attacks North east Nigerian city; dozens killed

The cut-up's darkly incongruous effect is achieved by replacing what we assume is Boko Haram with the mental image of a rampaging beauty in a swimsuit and a sash. But what was the other headline that sacrificed its Miss Universe to create this comic collision, and what if the bot had grafted Boko Haram into that headline instead? It's clear that some substitutions generate more humor than others, but the bot is blind to the differences. It sparks incongruities, but it cannot weigh their surprise value or guess at how appropriate they might be. Consider this example:

President Trump Trying to Bring Nintendo Entertainment System Back to Life

It is comforting to realize that this scarily plausible text is not an actual headline and that *President Trump* has been parachuted into a more banal headline in place of *Nintendo*. Ideally, the original entity and its replacement will be ontologically similar—for example, persons are replaced with persons and places with places—so that the cut-up can preserve the semantics of the original framing. But a transplanted peg should also stick out in its new hole to show us that it is far from home. In the words of Elliott Oring, this is what it means to be appropriate *and* incongruous.⁸ Nonetheless, an incongruity that seems appropriate to you can seem insensitive to others. When a bot cuts into the news to make incongruity leak out, its cut-ups are always vulnerable to the charge “*Too soon!*”

Look again at the cut-up that unleashed Miss Universe on northern Nigeria: Is it offensive to repurpose a real-life tragedy in this way, or is its subversion of beauty pageant clichés (“I will work for world peace!”) a welcome relief from what often seems like an endless news cycle of misery and death? It is in the nature of relief humor that we take relief wherever we find it, especially from topics that might otherwise make us feel anxious or confused. Yet few bots that are built to relieve are also designed to be funny, since attempts

at unscripted humor always run the risk of giving offence. Jonny Sun's @tinycarebot tweets its scripted advice on the hour, from "get some fresh air, please" to "please ask for help if you need it." More than 100,000 followers welcome its clockwork interventions every day as it scatters a gentle dusting of mindfulness across their time lines. Although Sun is a professional comedy writer, he didn't build his bot to be funny. His scripts keep the bot on a tight leash so that it can never drift from its benign mission.

Bots drift off message when they look to their users or potential targets for the raw materials of their tweets. Still, this kind of unpredictability allows a relief bot to be humorously incongruous in ways that speak to the specific concerns of real users. Consider @pentametron, a bot designed by Ranjit Bhatnagar to turn prosaic tweets into serendipitous poetry. Poetry is a state of mind, and it only takes a little rhythm to sway readers into treating a text's narrow concerns as deeper or more universal than they may otherwise seem. In keeping with its name, @pentametron favors iambic pentameter and looks for recent tweets of five feet, or iambs, of one stressed and one unstressed syllable apiece, as when, in sonnet 12, Shakespeare writes "And see the brave day sunk in hideous night." Any tweet of ten syllables can be read in this way, but Bhatnagar's bot goes one better: it finds and retweets pairs of ten syllable tweets that also have rhyming last syllables, as in this couplet:

Love is a very complex concept, sigh

I love the people I'm surrounded by.

It's a coincidence that each line mentions *love*, but a little rhyme can also go a long way in persuading readers to perceive intention where there is none. We expect poetry to beguile and sway us with its rhythms, so we dig deep to find a unifying truth. Friedrich Nietzsche, the great stylist of philosophy, noted that "the wisest of us occasionally becomes a fool for rhythm," beguiled as we are by its "divine skip, hop and jump."⁹ A text can just seem truer when it has rhythm. The researchers Matthew McGlone and Jessica Tofaghbakhsh dub this the *Keats heuristic*, after the last lines of the poet's "Ode on a Grecian Urn": "Beauty is truth, truth beauty, that is all / Ye know on earth, and all ye need to know."¹⁰ McGlone and Tofaghbakhsh gave their students obscure maxims to evaluate for understandability and

truthfulness. The maxims all rhymed internally, as in “woes unite foes,” but they threw an equal number of nonrhyming paraphrases into the mix too, such as “troubles unite enemies.” The students rated the paraphrases to be just as comprehensible as the originals, but they considered the rhyming maxims to be more truthful as carriers of meaning. To paraphrase Nietzsche, we are just suckers for rhythm and rhyme. The accidental union of unrelated texts can spark an incongruity, but it’s the Keats heuristic that nudges us to find reasons to see the incongruity as appropriate.

@pentametron’s rhyming couplets are likely to share an implicit topic on days when everyone seems to be talking about the same thing. On Super Bowl Sunday, for instance, American football provides the mood music for a great many tweets:

So far the @Superbowl commercials blow
Not gonna even watch the halftime show

The bot’s divine skip, hop, and jump also encourages us to interpret a tweet pair at the metalevel, bringing to bear cultural knowledge that the bot couldn’t possibly possess:

It's Groundhog day tomorrow, by the way
okay, okay, okay, okay, okay

Bots like @pentametron cast their own brand of sympathetic magic over language. We expect good poetry to be rhythmic, rhyming (by tradition, at least), textually coherent, and profoundly meaningful, but we can also be lulled into thinking that one quality, especially a superficial one like rhyme, is a guarantor of all the others. While the rhyme may not fool us, it encourages us to suspend disbelief so that we attribute ripples on the surface of language to the deeper tidal forces of meaning.

EDUCATED INSOLENCE

It takes smarts to be a smart-ass or, as Aristotle might put it, it takes an education of sorts to be wittily insolent. Notice how the bots we have surveyed use different kinds and sources of knowledge to achieve their humorous

effects. First, there is procedural knowledge. @pentametron, for example, knows how to count a tweet’s syllables and how to tell if two words rhyme, while @twoheadlines knows how to recognize the named entities in a headline. The second kind is shallow knowledge, which may be little more than raw data packaged in an expedient form. The shallowest of the bots, @stealthMountain, knows just two things: that “sneak *peek*” is a valid spelling, and that “sneak *peak*” is not. While the database of spoilers shared by @enjoyTheMovie is larger, it is no less shallow. The bot knows how to link the titles of certain films to certain strings, but it does not know what those strings mean. In the same way, @twoheadlines knows nothing of the meanings of its headlines, beyond the fact that certain words denote named entities that it can swap in or out. Yet as we will see in chapter 10, an entity recognizer can also give the bot category insights that meaningfully guide its actions. Knowing that a name denotes a male or a female, or a person, place, or product, allows the bot to only swap like for like. This is our third kind of knowledge: ontological knowledge.

High-concept bots needn’t do much to tickle our fancies, and we can often hand-code their knowledge in an afternoon. More knowledge requires more effort, but even this can be automated. We can write a program to scrape websites for movie spoilers and quotable lines, and for many other kinds of shallow knowledge too. Ideas for new bots frequently emerge from the serendipitous discovery of a large trove of easily scraped data on the Web. Given the data—for instance, a list of old proverbs, a dictionary of dream symbols, or a chronology of noteworthy events in history—we might imagine how we could use it in nonobvious ways, perhaps in combination with other sources to spark incongruity. The shallow knowledge that is harvested in this way might be creative in itself, such as one-line jokes or witty comparisons from across the Web. Other knowledge, especially of the ontological variety, is best assembled by hand if it is to be used by an AI system to make fine decisions about what is or isn’t an acceptable output. Let’s drill into one instance of each: a shallow Web-harvested database and a deeper handcrafted ontology.

Creative uses of language are rarely marked as such. Metaphors, for instance, carry no syntactic evidence of their figurative intent, so it is hard

for a machine to sift “kill him with kindness” from “kill him with poison.” Similes, however, are the exception. We often think of them as metaphor’s artless cousin because they come marked with an explicit *as* or *like*, but this only makes them easier to find. In fact, the shallow disposability of most similes means that we are more likely to invent new ones on the fly, leading to a wider array of descriptive inventions. Moreover, similes can be harvested at Web scale by using “as [X] as [Y]” as a query template. By using a long list of adjectival qualities from an electronic dictionary, a machine can send specific queries—such as “as *sad* as *” or “*cursing* like a *”—to a search engine and scrape any matching similes from the result set. A machine that does this for thousands of qualities can harvest tens of thousands of similes, and acquire many different ways of conveying these qualities in its own future texts.

This scale is a virtue, but it still gives us knowledge of the shallow kind. A bot may know that “a box of snakes” is a vivid image for an unpleasant situation, but it won’t understand why this should be so. Nonetheless, similes have other markers besides *as* or *like* that can reveal a speaker’s intent. For instance, similes that are prefixed with *about*, as in “about as welcome as a turd in a punch bowl,” are far more likely to be creative than those that are not. When authors wish to signal that a simile is exaggerated, playful, or ironic, a marker of semantic imprecision such as *about*, *almost*, or *not exactly* can fit the bill nicely.¹¹ As we’ll see in chapter 9, a machine can reliably use such markers to tell if a simile is intended ironically and can also use a simile’s syntactic structure to detect an incongruent pairing of ideas, such as “a chimp *in* a negligee,” “a wig *on* a dog,” or “a zombie *at* a dinner party.” Armed with this knowledge and with tens of thousands of vivid mental images at its disposal, a bot can decide when and how to use any particular one for humor.

Ontological knowledge is not so easily harvested from text corpora or the Web. Generic facts of the form *chickens are birds* or *Berlin is a city* can be extracted from semistructured sources such as Wikipedia, or from online ontologies that provide the same data in an explicitly logical form, such as DBpedia.org. However, vivid, detail-rich descriptions of the stuff that jokes are made of—scripts, frames, norms, and cultural icons in all their pomp and with all their flaws—are harder to come by. In the short term, manual

efforts can yield larger dividends than an automated solution for acquiring content like this. The NOC (nonofficial character list) is one such effort that my group has undertaken to support automated story generation. The list is so-named because much of its content is subjective, judgmental, and perhaps even slanderous if taken too literally. It is a source of talking points, not hard facts, for over a thousand cultural icons, from Cleopatra and Fred Flintstone to Hillary Clinton and Rocket Raccoon. For each of these icons, the NOC suggests positive and negative qualities, assays a political slant, lists any significant others, describes typical clothing and activities, suggests apt weapons and vehicles, and places each in its correct profession, genre, and domain. Fictional characters are additionally linked to their creators and to actors that have played them on stage or screen.

The NOC is a large collection of reusable frames, in the mold of those we met in chapter 1. In this pop culture LEGO set, the icons have the largest frames with the most slots, but the fillers of these slots also have frames of their own, to describe the paraphernalia and activities of different icons at lower levels of detail. We'll see how the NOC can be used for humorous storytelling in the next chapter, but for now, we treat the NOC as a box stuffed with all of our favorite toys. A bot can cut up its frame structures in much the same way that Burroughs and Gysin cut up newspapers, albeit with greater precision and more control over the outcomes. So a bot can put Homer Simpson into the White House or Donald Trump into space, marry Jack Sparrow to Lizzy Bennet, or have Maleficent star in *Sex and the City*. In all, the NOC contains thirty thousand cultural facts that allow stories, analogies, blends, and jokes to tap into our attitudes to familiar figures. It then becomes the task of a humor generator to combine these familiar cultural icons in playfully nonobvious ways.

READ ME LIKE A BOOK

@ReadMeLikeABot is a Twitterbot that uses its extensive knowledge of books to make personalized recommendations to those who tweet at it or who tag their tweets with its calling card, #ReadMeLikeABook. The bot was developed in 2018 for the annual Science and Communication

conference in Dublin, and while the event’s organizers tasked us with making the bot wittily provocative, its core functionality was well defined from the outset. First and foremost, it was to be a recommender of books to Twitter users on the basis of the user’s triggering tweet, or Twitter biography, or most recent posts. Given the aims of the conference, the bot was to recommend books on a scientific topic whenever possible, and it was to prioritize well-known—and still in print—texts over those from Amazon’s long tail, so that users might appreciate the wisdom of its choices. With its responsibilities framed in this way, we were set free to doodle around the edges with some additional flourishes of automated humor.

The bot taps into a variety of kinds and sources of knowledge, ranging from the shallow book knowledge it harvests from specialist sites, to the long list of comic similes it harvests from the Web, to the more structured cultural contents of the NOC.¹² @ReadMeLikeABot stores its knowledge of books in a layered database that it can later peel like an onion. At its core is a manually curated list of four hundred well-known books that reflect the knowledge and tastes of its creators. A secondary layer includes a further sixteen thousand books that have been automatically harvested from the Web ontology DBpedia.org. Each of the books in each layer is indexed on its title, its authors, and its various genres and topics so that it can be matched to the apparent tastes of a Twitter user. Books can also be matched to each other, so that the bot can recommend more of the same to eager readers.

The bot has a keen memory for what it has already suggested and strives for diversity in its recommendations. It won’t push the same book twice to the same user or make the same recommendations to different users on the same day. This fear of repeating itself can leave the bot with nothing to suggest, so it needs a third layer of books to delve into when it finds itself in a corner. While the bot doesn’t actually feel fear, this is precisely the kind of situation in which humans attempt humor. In this spirit, the bot’s outer layer is populated with nonce books that it invents for itself, to poke fun at the publishing industry’s silliest tropes. Whenever it backs itself into a corner, it uses jokes to laugh its way out.

Each fictitious book title must be sensible and nonsensical at the same time: silly enough to be seen for what it is, a joke, but sensible enough to

meaningfully relate to the user’s interests. Although each nonce book does not really exist, the tiny nugget of world knowledge that is reworked to give it its name must convey some understanding of real books and of the human condition more generally. As we’ve seen, one possible source of this knowledge is the wisdom of the Web, as found in linguistic forms such as the “about” simile. These are crammed full of familiar ideas in ways that tickle the imagination, making them a reliable source of appropriately incongruous word combinations. Consider the simile “about as conspicuous as a fart at a chili convention.” The adjective on the left may be playfully ironic, but the combination on the right is still strangely logical. What at first appears to be nonsense soon gives way to common sense, just as a classic murder mystery moves us from bafflement to revelation. As such, these strange combinations also suggest strange cases for a fictional detective to investigate:

The Odd Affair of the Fart at the Chili Convention: Lady Hawking Investigates

Detective Dexter and the Strange Case of the Zombie at the Dinner Party

The Baffling Affair of the Dog in the Sweater: An Inspector Grisham Mystery

Even when a combination is anything but incongruous, this lack of strangeness is itself a reason for curiosity, much like the dog that *didn’t bark* in the night:

The Mystery of the Corpse at the Funeral: Another Case for Dr. Chomsky

Three simple grammars are used here: one packages a simile as a mystery to be solved, another produces a new but believable name for the lead detective, and a third generates a name for the book’s author. The first two grammars tap into tropes of the mystery genre, while the third simply cuts up the names of existing authors to generate ephemera such as J. R. R. Hawking and Stephen Tolkien. Each book is then indexed on the ideas in its title, whether *chili*, *funerals*, *sweaters*, or indeed *farts*, so that the relevant book can be pulled out when the bot is in a jam.

The bot can make a serious recommendation of a real book, or a whimsical recommendation of a nonce book, but it can also make a half-serious suggestion of both. In this way, the appropriate incongruity of a nonce title

is heightened by its use in an otherwise serious context, as in this combined form:

Hey @███████████, given your personality profile I don't know which philosophical book on the solitude theme is more you: "Steppenwolfe" by Hermann Hesse, or "The Bowel Movement" by Stephen Tolkien.

If it's funny because it's true, our generators need generous helpings of received wisdom that can be subverted and repackaged in a joke format. But knowledge of an interpersonal nature—what we might call “gossip”—is thin on the ground in the existing databases that a generator might exploit. To remedy the lack, the bot uses the NOC that we met earlier. By blending a variety of NOC fields, the bot generates apt but fictitious book titles such as:

The Comedienne's Guide to Ranting about Liberals, by Roseanne Barr

The Rock Star's Guide to Avoiding Taxes, by Bono

The Psychiatrist's Guide to Probing the Mind, by Frasier Crane

Here the machine packages the NOC's profession slot with the typical activity slot to generate an imaginary but oddly appropriate book for a specific cultural icon. When this individual is also a fictional entity, the machine can instead attribute authorship of the book to whomever the NOC lists as their creator, as in:

Captain Ahab's Guide to Chasing a Great White Whale, by Herman Melville

Dr. Stephen Strange's Guide to Performing Magic Tricks, by Stan Lee

Yoda's Guide to Promoting Mysticism, by George Lucas

It can also suggest incongruous pairings of coauthors for the same book, as in:

The Geek's Guide to Studying Science, by Peter Parker and Wesley Crusher

The Son's Guide to Disappointing the Family, by Fredo Corleone and Bart Simpson

The CEO's Guide to Pioneering New Technologies, by Tony Stark and Steve Jobs

The bot recommends nonce books to a user only when real books from its inner layers cannot be found or when all have been offered to that

user already, so the value of its nonce inventions is largely revealed through repetition: the more a user interacts with the bot, the further from its core layer the bot must explore to find new titles to suggest, and the more the bot reveals of its sense of humor. Repetition of this kind fosters familiarity and trust, so it makes sense that it is at this point in its relationship with the user that the machine dares to be funny.

This kind of trust can also license the use of irony. Since the bot knows which of its similes have an ironic slant—these amount to over fifteen thousand similes—it can intentionally reshape these in ways that preserve but redirect the irony. In the following tweets, the bot pokes fun at Amazon’s recommendation algorithms while also tweaking the literary assumptions of its own followers:

Amazon customers who bought Marquis de Sade’s “The 120 Days of Sodom”
also bought these sensual titles:

Herpes, by Sady Grass

The Long Wet Kiss from Aunty Betty, by Gregor Morgan

How very sensual, indeed!

Amazon customers who bought Thomas Pynchon’s “Inherent Vice” also
bought these articulate titles:

The Lug Wrench, by Louis Lewis

The Untrained Chimp, by Paulo Raymond

How very articulate, indeed!

If the magic of a simile resides in its blend of mental images, then the reshaping process should limit itself to function words only, leaving the images conveyed by the nouns, verbs, and adjectives intact. This allows the bot to retain the spark of the original joke while adapting to the syntactic norms of its chosen target. Think of this reshaping as a blend that is conceptual *and* formal, insofar as it retains the concepts of one input but squeezes them into the form of another. New syntactic forms for preexisting combinations can be added quickly to adapt the bot’s stock of humorous similes to new fads. Suppose, for instance, that we want to mock the trend for quirkily

named brew pubs and craft beers. Traditional English taverns have rather formulaic names like *The Dog and Pony* or *The Ox and Cart*, so it is a challenge to see if our bot can show more imagination when inventing its own.¹³

To poke fun at old naming traditions and new pub trends, a Twitterbot named @OldSkoolFunBot uses all of the usual formulas but fills them with juxtapositions of ideas from its stock of Web similes. For “about” similes of the form “as [X] as [Y] <preposition> [Z]” it transfers the fillers [Y] and [Z] into the pub template “The [Y] and [Z]” in the expectation that the wit of the simile will persist in its new shape. Can you see yourself drinking in The Pork Chop and Synagogue, The Hippo and Tutu, or The Dog and Sweater? If so, what will you have to drink? To invent quirky new beers for its new pubs, the bot also turns similes of the form “as [X] as [Y][Z]” into “[X][Y] <beer>,” so that “friendly as a rabid dog” becomes Rabid Dog IPA and “firm as a wobbly jelly” becomes Wobbly Jelly Ale. As shown in the following tweets, the satire is fueled by ironic pretense and shaped by an echoing of syntax:

I'm off down to my local microbrew pub, The Drop and Bucket, for a pint of Medieval Ordeal ale.

Fancy going down to the new microbrewery, The Bug and Rug, for a pint of Hungry Snake weizenbeer?

Since the bot does not fully appreciate the intuitions behind the original similes, it can hardly claim to understand its repackaging of them. This introduces a degree of randomness that is sometimes serendipitous, as in the following tweets:

Fancy going down to the new microbrewery, The Dog and Wheelbarrow, for a pint of Golden Retriever lager?

I'm off down to my local microbrew pub, The Fart and Spacesuit, for a pint of Bottle Rocket ale.

The simile is just one of many linguistic containers of colorful mental imagery. If we take some care over our choice of templates, a simple bot can pour the wit from one into another without losing too much of the original's ineffable charm.

I HAVE THE BEST WORDS

If a bot's knowledge has a practical value to others, it can show off what it knows with an air of humorous superiority. If this knowledge is detailed enough, it can blend it in deliberately incongruous ways to make fun of itself or of cultural icons that are known to all. A bot can poke fun at famous people by inserting them into settings that seem both appropriate and incongruous, or it can attempt to mimic their distinctive voices. For instance, when a famous icon is already on Twitter, a statistical model can use its tweets as training data to attune to the regularities inherent in its use of words and syntax. Although its muse is now banished, a bot named @deepDrumpf used a recurrent neural network to acquire the word patterns and emphases of @realDonaldTrump's own tweets, so that it could then produce new ones of its own.¹⁴

@TrumpScuttleBot is one of many other Twitterbots that parody the tweets of US president Donald Trump.¹⁵ In its core operation, it caricatures the outbursts and obsessions of the man himself, using the word choices and framing devices of his own tweets. Incongruity arises naturally when familiar norms are squeezed into linguistic frames that a more thoughtful writer might consider inappropriate. Unlike @deepDrumpf, the bot does not use a statistical model of Trump's past tweets to generate its own, rather it uses explicit rules to frame its various knowledge sources in ways that make a virtue of inconsistency and incongruity. As a consequence, it does not learn from its target's tweets or acquire new topics and concerns that track the news and his role in it. The bot trades topicality for control, so that it can explicitly engineer the collision of ideas in ways that are likely to be humorous. The following is typical of @TrumpScuttleBot's brash and intemperate outputs:

If wardens don't criticize the way I run the country I won't slam the way they run their prisons, unless those prisons are about as secure as an old garden shed with a faulty lock #MAGA

Sarah Silverman is the latest actor to join the faculty at #TrumpUniversity's New Hampshire campus. As Professor of Stand-Up Comedy studies, she will give TREMENDOUS lectures on insulting minorities. Enroll now!

The bot combines its stock of Web similes with a stock of ontological norms, such as the expectation that prisons should be secure, to imbue its rants with a tangy vividness. But it can also combine these similes with expectations from the NOC:

Despite what those CNN liars tell you, Austin Powers is as hip as a hip replacement, BELIEVE ME!!

Despite what those transgender troops tell you, Boris Johnson is as invigorating as shopping at Old Navy with a 20 dollar spending limit, I should know!!

Despite what those Obama cronies tell you, Oscar Wilde is as colorful as a chest x-ray, BELIEVE ME!!

The bot also uses similes to respond to direct messages in ways that show it to be as sharp-tongued as it is thin-skinned. This response mode strives for superiority by producing zingers that relate directly to an incoming word. When it cannot do so, it falls back to the metalevel and criticizes the incoming tweet for not being topical, fascinating, deep, insightful, useful, or intelligent enough. The man himself is not given to this kind of ironic barb, but the bot aims for a deeper truth about its target's need for humorous supremacy, and his need to always have the last word:

From @ [REDACTED], replying to a tweet that also mentions @TrumpScuttleBot
→ Ties? He couldn't be further away.

From @TrumpScuttleBot, replying to @ [REDACTED]
→ Thanks for the useful feedback. It's as useful as a bag full of farts.

From @ [REDACTED], replying to @TrumpScuttleBot
→ Open it up and get a whiff . . . aaaaahhhhhh

The bot and its conversational partner quickly settle into a game of trumping, in which each strives to “own” and turn around the other’s metaphors.¹⁶ The bot does this with vivid similes that hinge on a word that the human has just used literally, but when it cannot find a match, it turns to a dictionary of dream symbols that it has harvested from the Web. These faux-profound insights lend the bot’s replies a condescending tone as it psychoanalyzes a human’s use of a specific word or idea:

→ Open it **up** and get a whiff . . . aaaaahhhhhh

From @TrumpScuttleBot, replying to @██████████

→ To tweet of being or moving **up** suggests you are emerging from a depressing or negative situation.

From @██████████, replying to @TrumpScuttleBot

→ Yes Dr. **Wizard**

From @TrumpScuttleBot, replying to @██████████

→ To mention a **wizard** in your tweet suggests you are trying to hone your skills and exercise your power.

From @██████████, replying to @TrumpScuttleBot

→ OK **Goofy**, if it makes you feel better.

From @TrumpScuttleBot, replying to @██████████

→ You make Jar Jar Binks seem about as **goofy** as doomsday.

From @██████████, replying to @TrumpScuttleBot

→ **Good** one, you got me.

From @TrumpScuttleBot, replying to @██████████

→ You seem about as **good** as a movie about a leprechaun in space.

From @██████████, replying to @TrumpScuttleBot

→ At least Trump is president.

From @TrumpScuttleBot, replying to @██████████

→ Thanks for the useful feedback. It is as useful as a knock on the **head**.

From @██████████, replying to @TrumpScuttleBot

→ You keep responding, so who's **head** is being knocked?

From @TrumpScuttleBot, replying to @██████████

→ To mention a **head** in your tweet signifies wisdom, intellect, understanding, and rationality.

These strategies ensure that @TrumpScuttleBot always has the last word. It may seem incongruous that a political satire bot spends its spare time indulging in the psychobabble of dream interpretation, but it's the nature of comedy bots to seek out incongruity, and the nature of humans to ride some surprising hobbyhorses.

HAVE IT YOUR WAY

Many bots are simple enough, or high-concept enough, to wear their generative processes on their sleeves. This is just one of the reasons so many of us knowingly follow them on Twitter: we appreciate their outputs, warts and all, when we know how they were generated. If a bot is a self-contained system of frames and fillers that must be combined in particular ways, we can define its generation processes as a semantic grammar. Elegant formalisms such as Tracery will then allow us to develop a bot as a set of production rules with little or no programming.¹⁷

We can think of a semantic grammar as a set of rules that govern the structure and content of the bot's outputs. Unlike a grammar for English, say, in which the rules of sentence formation are defined in terms of syntactic categories like *noun*, *verb*, and *adjective*, a semantic grammar defines its rules via ontological categories that denote things of interest in its domain, such as *person*, *weapon*, or *vehicle*. Consider how we might define a simple semantic grammar for generating a pizza:

```
<pizza>      → <toppings> with <cheese> and <sauce> on <base>
<toppings>   → <topping>
<toppings>   → <topping> and <toppings>
<topping>    → <meat>, <veggie>
<cheese>    → "mozzarella", "cheddar", "provolone", "Romano"
<sauce>     → "marinara sauce", "barbecue sauce", "salsa"
<base>       → "thin crust", "thick crust", "flatbread"
<meat>       → "ham", "sausage", "pepperoni"
<veggie>    → "peppers", "olives", "onions", "sweet corn"
```

In this formulation, our ontological categories are placed in angle brackets, and the instances that populate these categories are given in quotation marks. Each rule has its defining category on the left of the arrow and one or more expansions on the right. Notice how the <meat> rule expands as a set of possible instances of the meat category. Since a comma denotes choice, this rule can be understood as follows: to produce a <meat>, choose one of "ham" or "sausage" or "pepperoni." In the same vein, the <pizza>

rule can be understood as follows: to produce a `<pizza>`, first produce its `<toppings>`, then its `<cheese>`, then its `<sauce>`, and then its `<base>`, where each of those other rules is expanded before all of the results are finally glued together. So when we ask our grammar to generate a new `<pizza>`, this is the kind of output that we can expect: “ham and sweet corn with Romano and marinara sauce on thin crust.”

Designed by Kate Compton, Tracery is an expressive framework that uses JavaScript Object Notation (JSON) to express rules like these. Our pizza grammar can be carried across wholesale:

```
{  
    "origin": ["#pizza#"],  
    "pizza": ["#toppings# with #cheese# and #sauce#  
              on #base#"],  
    "toppings": ["#topping#", "#topping# and  
                 #toppings#"],  
    "topping": ["#meat#", "#veggie#"],  
    "cheese": ["mozzarella", "cheddar", "provolone",  
               "Romano"],  
    "sauce": ["marinara sauce", "barbecue sauce",  
              "salsa"],  
    "base": ["thin crust", "thick crust",  
             "flatbread"],  
    "meat": ["ham", "sausage", "pepperoni"],  
    "veggie": ["peppers", "olives", "onions",  
               "sweet corn"]  
}
```

Tracery’s JSON syntax imposes some superficial changes on our grammar, but its shape remains unchanged. Whenever ontological categories are mentioned on the right-hand side of a rule, we wrap them in `# ... #` to indicate their special status. This conveniently allows categories to be expanded within a wrapper of literal content. We also need a special

category, the “origin,” to indicate the entry point to the grammar. It is this category that the generator expands when it is activated.

CheapBotsDoneQuick.com is a bot-hosting platform that makes it easy to deploy your Tracery grammars as Twitterbots. Built and managed by George Buckenham, the site authorizes your Twitter account for use as a bot and allows you to enter a Tracery grammar to define its generative reach. You can then schedule your bot to tweet the outputs of its “origin” rule at intervals of your choosing. If you wish, you can also specify an additional response grammar that allows your bot to respond to any direct messages or any mentions of its handle in other people’s tweets.

A response grammar is simply a mapping of input strings to output strings, but the latter may contain ontological categories from the main grammar. So when an incoming message contains a specific trigger phrase, the corresponding response is issued as a reply, after any categories have been expanded in place. If your pizza grammar contains a category for “vegan pizza,” your response grammar may link the trigger phrase “vegan” to this response: “I made you a vegan #vegan pizza#.”

Twitterbots with fine-grained categories are more discriminating in what they generate and in how they reply to requests or provocations from others. But how can we squeeze humor into the rules of a bot as simple as a pizza generator? Our pizzas are random combinations of ingredients from different categories, but this randomness is tempered by some top-down forces, such as our vegan rule. In this sense, a Tracery grammar is an ontology-driven application of the cut-up method. If our bot can meaningfully name the pizzas it generates, this will further burnish its bona fides as a producer of creative content. Consider this pizza from a Tracery bot whose grammar builds significantly on the one given above:

I made this pizza with cod goujons and baby spinach and called it “Popeye’s Full Monty” because it also has Monterey Jack on hummus on wholemeal crust.

Notice that the name of the pizza is given in the middle of its description, since the bot uses a linguistic principle known as center embedding to simplify its grammar. This allows a pizza to be generated as two halves that stitch together at the name, so each name must also comprise two

parts: the first refers to an ingredient in the first half of the pizza and the second to an ingredient in the remainder. In this case, “Popeye’s” is a name prefix associated with the ingredient “spinach,” while “Full Monty” is a suffix associated with “Monterey Jack.” The combined name is a surprising but apt cut-up of images that emerges out of the pizza itself.

We can make our grammar as fine-grained as we like and add as many cultural associations to its naming rules as we can think of. We might associate “Swedish meatballs” with “Abba” and “IKEA” and link “Kobe beef” with “Godzilla,” so that these clichés can later collide with others in a pizza that makes them appropriate *and* incongruous as a name. We can even augment our response grammar to suggest wines to drink with the bot’s pizza inventions or have our pizza bot delegate to a Tracery wine bot (via a mention of its handle) to supply the recommendation. Small grammars can grow in size and ontological sophistication to provide some surprising functionality, and each new add-on provides further scope for humor. Why not experiment with these pizza grammars for yourself in our repository?¹⁸

To the extent it is humorous, our pizza bot is a humor system with a day job. It exists to generate pizzas, but it uses the concerns of its domain to also be funny. So let’s consider a bot whose only goal is humor. As we have seen, comedy owes as much to the juxtaposition and careful framing of familiar ideas as it does to the invention of brand new forms, and comedians have such large stocks of familiar surprises to draw on. Yet so do our machines, in the guise of tens of thousands of humorous similes that can be harvested from the Web. We can write a program to coalesce this army of similes into a legion of Tracery rules, one rule per quality:

```
+subtle": ["irony", "a watercolor", "a lover",
            "a breeze", "a smile", "a whisper",
            "a sigh", "a ninja", "a ballerina",
            "a butterfly", "perfume", "intuition",
            "a watermark", "eye contact", "an orchid",
            "a snake", "a pickpocket"],

"-subtle": ["a brick", "an army tank", "a sledgehammer
            to the temple", "a neon sign", "a
```

```
jackhammer", "a freight train", "a strong  
kick in the head", "a hand grenade", "a  
punch in the face", "a train wreck", "a  
spear through the head", "an elephant", "a  
tank", "a punch in the neck", "a kick in  
the face", "a visit from the goon squad",  
"a steamroller", "an atom bomb", "a lead  
pipe", "a brick to the face", "the can  
can", "an electric shock", "a kick in the  
pants", "a supernova", "a two-by-four to  
the head", "a falling brick", "a shotgun  
blast", . . . ]
```

We use + and – to mark the straight and ironic uses of a quality. So to embed an ironic counterimage to “subtlety” in a tweet, a grammar need only reference the category #‐subtle# in a rule. Note that the rule for “‐subtle” offers a choice of 215 ironic expansions, many of which allude to attacks on the head or neck, so just a small sample is listed here. In all, the grammar defines 754 ironic rules like this and 732 nonironic rules. By cutting and pasting these rules into a bot’s grammar, we can make this trove of exaggerated or subversive images available for it to use.

To see how, let’s build a simple bot that makes sweeping generalizations about different countries. The following shows all but the simile rules of the grammar:

```
"origin": ["1000 metaphors for Trump's America:  
 \n. #America#",  
 "1000 metaphors for Putin's Russia:  
 \n. #Russia#"],  
 "America": [#‐subtle#, #-appealing#,  
 "#‐welcome#"],  
 "Russia": [#‐ruthless#, #‐threatening#,  
 "#‐vicious#"],  
 "number": ["Number #digit1##digit1##digit2#"] ,
```

```
    "digit1": ["", "1", "2", "3", "4", "5", "6",
               "7", "8", "9"],

    "digit2": ["0", "1", "2", "3", "4", "5", "6",
               "7", "8", "9"]

}
```

This grammar is a basic exercise in framing, one designed to squeeze an existing resource into new forms that make its inherent humor more pointedly topical. Consider these outputs, which tap into the categories “–welcome” and “+ruthless”:

1000 metaphors for Trump's America:

Number 269. Subpoenas

1000 metaphors for Putin's Russia:

Number 537. The Gangster

Neither of these exercises in controlled randomness may seem very topical by the time you read this, but that is largely the point of agile grammar development. By framing the categories of an existing ontology in different ways to suit our current goals and inclinations, bot grammars allow us to move quickly with the times. So why not try your hand at your own framings of the grammar's legion of similes? Just download the grammar from the repository and see what works best for you.

NOC, NOC

The call–response structure of a knock-knock joke makes it an ideal showcase for humor through bot interaction. As when kids tell this kind of joke, one bot must initiate the formula by saying, “Knock knock,” and the other will reply with, “Who's there?” In the NOC-NOC subspecies of the joke we explored earlier, the first will then respond with the first name of a cultural icon, such as “Kanye” or “Kim,” and the second will ask for a surname. At this point, the first bot will engage one of its script templates and fill it with details from a NOC entry with the given name.

Let's call the first Twitterbot `@NocNocBot` and the second `@NoccerBotter`. The main Tracery grammar for the first bot is very simple and has just a single rule:

```
{  
  "origin": ["Knock, Knock @NoccerBotter"]  
}
```

We now use CheapBotsDoneQuick.com to schedule the bot's knocks on our desired timetable, such as once every six hours. This first bot's response grammar must also be configured to accept the returning “Who's there?” from `@NoccerBotter`:

```
{  
  "Who's there?": "#firstname#"
```

Notice that a response grammar allows only one expansion on the right-hand side, but this expansion can refer to any rule in the main grammar. We are not going to close out this grammar yet, as we will add more response capabilities later. As for `@NoccerBotter`, we need it to respond to any well-formed input, from us or from its partner in comedy, so its response grammar must handle any NOC first name:

```
{  
  "Knock, Knock": "Who's there?",  
  "Bill": "Bill who?",  
  "Hillary": "Hillary who?",
```

The response grammar of `@NocNocBot` must now handle these replies as follows:

```
"Bill who?": "#Bill_response#",  
"Hillary who?": "#Hillary_response#",
```

A rule like “Bill_response” which will be defined in the main Tracery grammar of @NocNocBot, acts as a clearinghouse for all NOC characters named “Bill”:

```
"Bill_response": ["#Bill_Clinton#",  
                  "#Bill_Cosby#", "#Bill_Gates#",  
                  "#Bill_Murray#",  
                  "#Bill_O'Reilly#"],
```

Each of these specific expansions will in turn generate responses that flesh out a NOC-NOC template with a different aspect of a character’s entry in the NOC. These aspects may be shared by several NOC characters, becoming rules in themselves:

```
"Bill_Clinton": ["Bill Clinton. #spouse_Hillary#",  
                 "Bill Clinton. #namesake_  
                 Bill_O'Reilly#", "Bill Clinton.  
                 #namesake_Bill_Cosby#", . . . ],  
  
"namesake_Bill_O'Reilly": ["Bill O'Reilly is busy  
                           'promoting conservative  
                           values' with your  
                           momma", . . . ],
```

The idea of writing thousands of rules for the main and response grammars of each bot may strike you as a Herculean task, but there is no need to do this by hand. We can simply write a *generator-generator* to do it for us. This piece of code takes the NOC and our list of templates and turns their crossover possibilities into two main grammars and two response grammars. This is more than the automation of an onerous task, since it generates complex grammars that can surprise even the creators of the data on which they are based. When we paste these grammars into CheapBots-DoneQuick.com, our bot duo will now produce interactions like these:

@NocNocBot: Knock, Knock @NoccerBotter

@NoccerBotter: Who’s there?

@NocNocBot: Leonardo.

@NoccerBotter: Leonardo who?

@NocNocBot: Leonardo da Vinci. Do you mind if I park my scythed chariot in your driveway?

As you can see, Tracery is an elegant framework in which to build an idea-driven generator of content. It provides a wealth of handy features that I have not covered here. For its part, CheapBotsDoneQuick.com makes deploying and managing your Tracery bots a snap. You can build your bot grammars by hand, but as they grow in scale and complexity, you may find it more practical to write computer programs that write your grammars for you. George Buckenham wisely designed his hosting site to accept large grammars that are much bigger than any human would write.

The limits of bot building with Tracery and CheapBotsDoneQuick—or, indeed, with superficial cut-ups of open-world forms—are also, in a way, the limits of a weak engineering approach to implementing a theory of humor. There is only so much we can achieve in a framework for combining nuggets of meaning in ways that we hope, but can never quite guarantee, will yield meaningful or humorous outcomes. Although the surfaces of language can be remarkably revealing about what is going on beneath, we need our machines to dig deeper into the lower layers of meaning and intentionality if we want them to be fluent in the causes of humor. Nonetheless, our tour of basic generators in this chapter gives us a simple baseline for assessing the fluency of a humorous AI system, and in the chapters to come, we'll turn our gaze toward more complex models for the treatment of humorous words and actions. To combine the control of closed-world rules with the scope of open-world forms, we need to become more mathematical in our approach. However, as we explore bottom-up, statistical notions of meaning, we will keep an open mind as to the value of top-down structures in shaping meaning from above.

If a bot is just a robot without a body, then a physical robot is an embodied bot. In the next chapter, we will up the ante on the development of humorous bots by giving them real bodies in which to perform. The possibility that complex and potentially humorous behaviors might arise from the interactions of simple bots has been explored here in the context of a

pizza bot that defers to a wine bot to obtain high-culture recommendations for its low-culture inventions, and in the context of a pair of joke bots that play different roles in a call–response formula. But let’s see how far we can take this idea of an automated double-act when our bots have real bodies and a real stage on which to act out their comedic interactions.

4 DOUBLE TROUBLE: HUMOROUS STORYTELLING AND EMBODIED AI

CHEESE AND CRACKERS

Perhaps you've heard the joke about the disgruntled employee who asks his boss for a raise. "I do the work of two people for this company," he complains. "I know," sighs his boss, "*Laurel and Hardy*." There are just some people you should never ask to move a piano, because less really is more when coworkers undermine rather than support each other. But comedy turns this conventional wisdom on its head. We love the ill-matched pairings that teeter on a knife edge, the duos that fight like cats and dogs, and the couples that go together like a pair of old boots, even if each boot is aimed squarely at the other's head. Theirs is a special kind of friction.

Comedy partnerships have their roots in the world of Victorian music halls and the burlesque and vaudeville stages, where comics of opposing temperament saw that they could elicit more laughs performing together than alone. The first duo to successfully make the leap from silent to talking pictures, Stan Laurel and Oliver Hardy, is still considered the high-water mark of the form.¹ Stan & Ollie differed in obvious ways—they looked different, spoke differently, thought differently, and used their own signature mannerisms—but their friendship persisted. Ollie was not a conventional straight man, as he was neither smart nor suave, but he did act as though he were the brains of the partnership, and his reactions to an incongruous situation or to his partner's idiocy anticipated the audience's own.

The line between straight and funny man was more sharply drawn in pairings such as Lou Abbott & Bud Costello or Dean Martin & Jerry

Lewis, and it's here that the straight man comes into his own. As his zany partner frantically paints outside the lines, the straight man places a rigid frame around the emerging picture. As lightning rod to his partner's lightning, the straight man must ground the comedy while stealing little of the funny man's thunder. Although traditional comedy duos are now the stuff of TV reruns, theirs is a dynamic that still thrives, in cartoon friendships from Fred Flintstone & Barney Rubble to SpongeBob SquarePants & Squidward Tentacles; in sci-fi pairings from Dr. Smith & B-9 to C-3PO & R2-D2 to Cooper & TARS; in TV morning shows; and even in the way the news is reported.

If the chemistry of a pairing is compelling enough, it can transcend the oddest of technological barriers, as in the pairing of Edgar Bergen and Charlie McCarthy.² Bergen was an American ventriloquist in the 1930s and 1940s who gave his dummy Charlie the freckles, red hair, and feisty personality of a real Irish newspaper boy. Bergen had studied with the best to become a consummate stage ventriloquist, so it is no small irony that he and Charlie would reach their biggest audiences on the radio. As amazing as it sounds now (Who would ever tune in to hear a magician do card tricks on the radio?), millions avidly tuned in to *The Chase and Sanborn Hour* to hear Charlie subvert Bergen's suave paternalism with his own brand of impish wit. Listeners didn't tune in to appreciate Bergen's vocal technique, for who could know if he ever moved his lips or not? Rather, it was the duo's obvious chemistry that persuaded listeners to move the dial, and the volatile clash of opposites, not least the clash of the real and the imagined, that kept them coming back for more.

The popularity of the traditional ventriloquism act shows that audiences really don't mind if one of the performers in a comedic double act isn't entirely human. Ventriloquists have made stuffed ostriches, foam emus, and tinfoil robots come to life, so it isn't a stretch to imagine a double act of a comedian and a real robot, or a comedian and a laptop running a chatbot. AI has a long, if not always illustrious, history with chatbots. The first and most famous was *Eliza*, Joseph Weizenbaum's cautionary tale of a bot that shows just how much a pinch of syntactic trickery can do to further a dialogue and suggest the impression of an intelligent interlocutor.³

Weizenbaum named his system after the upwardly mobile flower seller of George Bernard Shaw's *Pygmalion*, but we remember it most for its specific instantiation, the *Doctor* script, in which the system acts the part of a Rogerian therapist. In this role, Eliza is scripted to rephrase, repackage, and deflect the user's thoughts while contributing none of its own to a dialogue. As a conversational free rider, Eliza is most effective when users allow its scripted responses to draw them out on their own concerns. The bot's great trick is to turn users into unwitting ventriloquists that animate the apparently intelligent system with their own words and feelings.

If Eliza is a Flintstones car that needs a scurrying occupant to propel it forward, then Parry is a chatbot with an engine of its own.⁴ Designed by Kenneth Colby to model the inner thought processes of a paranoid schizophrenic, Parry takes the role of patient, while the human interlocutor plays doctor. In this important sense, Parry is more than a bag of syntactic tricks that is designed to keep a conversation moving. It is, in fact, a testable cognitive model of the paranoid personality. This puts the onus on Parry to contribute ideas to every conversation and flesh out the bones of an interlocutor's questions with meaty details from its imagined past. Colby based Parry on a real patient with a fascinating backstory involving horses, gambling, and a fear of organized crime, and so this is Parry's world too. He needs little excuse to tell you of his love of horse racing or his recent outings to the track.

But Parry's dialogue manager does more than find a response for every input. It constantly updates the hidden state variables that track Parry's internal levels of fear, threat awareness, and shame. While Parry's sense of shame may be little more than an internal counter, one that waxes and wanes with the conversation as other connected states undergo changes too, Colby puts it to some surprisingly sophisticated uses. For instance, if Parry takes too long to respond to a question, this embarrassment leads to an increase in its shame level. As this level rises, much like a levy about to burst, Parry's responses show mounting hostility as it projects its shame outward as an attitude of fear or contempt for the human user.

It is easy to imagine how an AI-literate standup might build a routine around a dialogue with Eliza or Parry.⁵ Colby's simulation of a combative

personality was designed to spark conflicts with a human interlocutor, so Parry naturally provides a wealth of material from which a comedian can eke out appropriate incongruity, although sensitivity is needed to ensure that audiences laugh *with* Parry rather than *at* the mentally ill. Eliza, in contrast, avoids conflict by design. Because the bot routinely deflects a user's inputs to maintain its clinical distance, it exhibits a level of predictability that makes it better suited to the role of straight man in a double act, perhaps to play Edgar Bergen to Parry's Charlie McCarthy.

Rather than just imagine Eliza in the role of straight man or Parry in the role of funny man, let's see what they actually do when we hook them up to each other. In 1973, just a year after Parry's development (Colby wouldn't publish his seminal paper on Parry until 1981⁶), the network pioneer Vint Cerf decided that a conversation between Eliza and Parry on the ARPANET—the earliest incarnation of what we now call the Internet—would be a fun way of showcasing the ability of remote computers to communicate as if they were sitting in the same room.⁷ It was a historic moment, to be sure, but as face-offs go, it was more *Frost/Nixon* than the *Thrilla in Manila* or *Rumble in the Jungle*. Each makes a predictable opening move:

Eliza: Good Evening. Tell me your problems. Please terminate input with a period or a question mark.

Parry: People get on my nerves sometimes.

Perhaps chess offers a more apt metaphor for this digital face-off? Even so, it's not always clear that both systems are playing on the same board:

Eliza: I am not sure I understand you fully.

Parry: You should pay more attention.

Eliza: Suppose you should pay more attention.

Parry: You're entitled to your own opinion.

Eliza: What makes you think I am entitled to my own opinion?

It takes Parry's ego fixation to nudge things back onto the same track:

Parry: It bothers me just to be around people in general.

Eliza: Please go on.

Parry: I went to the track at Bay Meadows a while back.

Eliza: What does that suggest to you?

Parry: I went to the races.

To a system like Eliza, a cigar is always just a cigar, and its phony (and phoned-in) interest in the symbolism of our actions is just a back-off strategy for dealing with content it cannot otherwise grasp. While this gambit often works well on humans, it is unlikely to get very far with a partner as literal-minded as Parry.

Eliza: What else would you like to discuss?

Parry: Horse racing interests me.

Eliza: Why do you say that just now?

Parry: There's a lot of crooked gambling going on there.

Eliza: I am not sure I understand you fully.

Sometimes we just want to open up about the one thing that means more to us than anything else in the world, but cigars, horses, gambling, and crooks are all the same to Eliza. Parry may be a fake too, but it's not far from the mark when it says:

Parry: It's clear to me if not to you.

Eliza owes its glibness to a cheap bag of tricks that hint at a complexity that just doesn't exist beneath the hood. But the effect can be as fleeting as it is fragile:

Eliza: Do you think it's likely that not to I?

Eliza converts the "you" of Parry's jibe into an "I," but the trick still falls flat here. However, Parry has a one-racetrack mind and only cares about its own grammar:

Parry: I went to the races.

Eliza: Please go on.

Parry: I like to gamble.

Eliza: What does that suggest to you?

Parry: I went to the races.

Eliza: What else would you like to discuss?

Neither bot comes out of this Beckettian *Waiting for Robo* farce covered in glory, but we can appreciate why Parry is so given to anger. Only Parry assigns any real content to content words. His model defines procedural and emotional reactions for different trigger words, while Eliza's is entirely content-free. Before systems like these can truly converse, we must first give them a shared ontology of how words describe the world so they can use them in mutually intelligible ways.

CHICKEN SOUP FOR THE ROBOT SOUL

Our interacting systems need to share more than a domain model that tells them about the things they can talk about or the words they should use when they do. They must also share a metamodel about the nature of conversation itself. You may have heard the maxim, often attributed to the journalist Miles Kingston, that says, “*Knowledge* is knowing that a tomato is a fruit, but *wisdom* is knowing not to put it in a fruit salad.” A domain model contains the first kind of information, the kind we find in dictionaries, encyclopedias, and the ontologies of AI. A metamodel is tasked with storing the latter kind, called wisdom, so that agents can use their knowledge in ways that, if not always wise, do not cause others to question their sanity. We each possess a staggering amount of propositional knowledge, but this is relatively straightforward to encode as logical axioms. Wisdom, however, is tacit and difficult to pin down in explicit rules. So, like Kingston’s aphorism, it must be captured in maxims that lack the weight of rules.

The simplest metamodel is always just a script. If each actor simply does *what* the script says they should do, *when* they are supposed do it, the meaning of their utterances will be reflected in their reactions. But if giving

machines handcrafted scripts smacks a little too much of old-timey ventriloquism, a solution is to have a machine write its own. A single machine can write story scripts for itself, or for other machines, or even for humans, in much the same way that professional writers create content for other performers. As they execute the script in tandem, each of the actors will be guided by the same domain model and metamodel, whether they are humans or robots. And as in the case of humans, the robots will also be free to ad-lib, do double-takes, or add commentaries of their own. They can choose to slavishly follow the script or break the fourth wall whenever it suits.

Scriptwriting focuses as much on performance as on plotting. A story is a thing, but a script is an event waiting to happen. Nonetheless, the generation of stories as textual artifacts has a long provenance in AI, and some of the most famous tale spinners are as old as Eliza and Parry. We can even view Parry as an interactive fiction, where the aim of our exploration is not to find treasure or slay a dragon but to get to the heart of Parry's psychosis.⁸ Just think of Parry as a text adventure game that reveals more of its protagonist's world the more that we interact with it. Yet no matter how interesting his story, Parry has only one tale to tell, and we soon exhaust its potential to surprise once we make return visits to its world of horse racing, gambling, and crooks. In contrast, even the simplest story generator explores a whole space of stories and is always capable of cranking out new ones.

James Meehan's Tale-Spin system from 1976 spins its tales of talking animals by first giving its characters some basic goals, such as finding juicy berries to eat, and by then giving them obstacles to overcome on the way.⁹ Scott R. Turner began work on his Minstrel system in the 1980s as a means of weaving Arthurian tales of courtly knights.¹⁰ Like Tale-Spin, it motivates characters by giving them goals to achieve and plans to follow. Minstrel explores a space of story possibilities that is defined by a mix of genre constraints and script-like plot structures to generate tales that are novel but never *too* novel. If Parry's hard-coded tale of woe were to be subcontracted to a generative system such as Tale-Spin or Minstrel, it could spin up a brand-new backstory, and a whole new reason to be paranoid, for each new user or for each new session. Parry would not then be fated to fall into a conversational death spiral whenever it interacted with a system like Eliza.

Eliza is no storyteller, but it must still keep a dialogue going and maintain the semblance of a narrative thread in the minds of its users. It does this by resorting to the shallowest of linguistic tricks. Eliza is a distant reader at the best of times and uses single keywords rather than complex constructions to trigger its specific responses. In many cases, it is a black mirror, reflecting back at users only what they themselves bring to the conversation. Yet even then, Eliza may appear to be reflecting more than just words and can even seem to be mirroring one's own intelligence back at the user. It's no wonder that Eliza occasionally beguiles its more sophisticated users too, at least for a spell, into assuming there is more going on than simple parlor tricks. Since few AI systems have done so much with so little, this illusion has come to be known as the *Eliza effect*. Douglas Hofstadter, who coined the term,¹¹ fears that the effect is too deeply ingrained in our relationship with machines to ever eradicate, while others, including Eliza's creator, Joseph Weizenbaum, have described it not so much as an illusion but a delusion.¹²

Conversely, when a more sophisticated system grinds its gears and exposes its failings, we become disillusioned by what Noah Wardrip-Fruin calls the *Tale-Spin effect*.¹³ A truly generative system, such as Tale-Spin or Minstrel, doesn't merely reflect the user's own intelligence but aims to simulate its own. It's a factory, not a recycling center. Should you ever find a hair or a fingernail in your hot dog, you will fathom more about the workings of a sausage factory than you ever wanted to know. Likewise, Tale-Spin often generates what Meehan dubbed "misspun" tales—stories with faulty logic or inane plots that bring the system's grasp of not just narrative but of words and meanings into question.¹⁴ In one misspun tale, a bear asks a bird for honey and is pointed to the nearest beehive. But the bear angrily takes the bird's response as a rejection—I imagine it hollering, "Show me the HONEY!"—because, unlike the bird, it fails to see any link between beehives and honey. At times like this, Tale-Spin's uneven complexity works against it.

The effect surfaced again when Meehan decided to add gravity as an agent in its own right. In Tale-Spin's causal model, characters and props need a reason to fall, and a personification of gravity is as good a reason as any. However, a subtle mover like gravity can become painfully visible when we least expect it. Consider this memorably misspun tale: "Henry

Squirrel was thirsty. He walked over to the riverbank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. Gravity drowned.” It was gravity that dragged Henry into the water, but at least Henry had Bill to save him. Invisible as it was, gravity could only drown.

Tale-Spin wasn’t designed to generate comic stories, but Meehan saw reasons to laugh in its misspun tales. He went large with its missteps, posting them in the corridors for the amusement of other researchers. Who wouldn’t laugh when a magician trips over his cape, sits on his top hat, or is pooped on by his doves? A deft performer might play along and milk the goofs for laughs. Better ones might encourage just this kind of humorous accident and make it a central part of their act. But systems like Tale-Spin lack the ability to see themselves as users do. After all, if Tale-Spin could see itself from the metalevel, it would never misspin a tale. In deciding which of its fails were funny enough to post, Meehan became his own system’s metalevel. But there is no reason why another AI system couldn’t fill this role. As one system generates, another can critique from outside. As cooperative rivals, they can work together to merely seem disharmonious, just as in a human double act. They can explore the space of comedic interactions in real time, acting as they go, or they can do it offline, to create scripts to be followed at a later time.

SILLY AND CHARYBDIS

As they navigate this space and make their choices, our double act of AI partners must steer a course through the jaws of the Eliza and Tale-Spin effects. With a suitable choice of framing devices, a partnership can profit from the Eliza effect when it talks of things that are beyond its ken, but it can also exploit the Tale-Spin effect if it packages its own shortcomings as comedic events. Indeed, even if it does not fully grasp the consequences of its failings, the Eliza effect can still add to how an AI harnesses the Tale-Spin effect to earn it more laughs than it deserves. Despite seeming like natural antagonists, the effects work rather well together.

It is possible to detect obvious goofs with simple heuristics. Consider another misspun tale about Joe Bear and Irving Bird. Joe still wants that

honey, but this time he offers a quid pro quo for the inside scoop: “He asked Irving Bird where some honey was. Irving refused to tell him, so Joe offered to bring him a worm if Irving told him where some honey was. Irving agreed. But Joe didn’t know where any worms were, so he asked Irving, who refused to say. So Joe offered to bring him a worm if Irving told him where a worm was. Irving agreed. But Joe didn’t know where any worms were, so he asked Irving, who refused to say. So Joe offered to bring him a worm . . .” When a generator gets stuck in a loop, it takes a metalevel viewpoint to nudge it back on track. No matter how sophisticated it may be, the AI will chase its tail at the domain level until the metalevel steps in.

Consider the cutting-edge approach to text generation that is OpenAI’s GPT neural framework. Their GPT-2 system,¹⁵ a neural language model trained on billions of words of Web text, made waves in the media when its creators expressed concerns that the largest of their statistical models posed too great a threat to public discourse to be openly shared.¹⁶ The system was just too good at generating believable texts, even for unbelievable topics such as the discovery of unicorns in the Andes, that it might be exploited by dark forces to automate the creation of fake news. A language model assigns a probability to every possible word sequence in a language, and so, for a given prompt, it can suggest the words that are most likely to come next. This allows a model based on GPT to accept a writing prompt and hallucinate the text that flows most naturally along the same groove.

To generate a lucid continuation to a prompt, *beam search* is typically used to find the word sequences with the highest likelihood of coming next.¹⁷ The beam heuristic constrains the avenues that the model is willing to pursue, but it can cause a model to favor the same high-probability continuations over and over again, as if stuck in a dream from which it cannot wake itself. Take this continuation from a GPT model with more than one hundred million parameters: “. . . looks at the clouds. He looks at the clouds . . .”¹⁸ Alternatives to beam search are better able to handle these snags, but the wider issue remains: even sophisticated generators can produce outputs that an outsider with a less sophisticated metamodel can recognize as defective.

Problems like these are easily predictable with hindsight, which is to say that a metalevel treatment is straightforward to implement once the problem has been observed during testing. It is then a simple matter for one partner to look for any unresolved references, inane repetitions, or other goofs in the outputs of another and to reply with an appropriate speech act, such as, “Well, don’t let me interrupt,” for a repetition, or, “Who the hell is ‘Gravity?’” for a dangling character reference. Each speech act serves a dual purpose, much as it does in human communication. Each assures the audience that the problem is under control and even suggests, via the Eliza effect, that it may be a deliberate part of the act. Moreover, each is a signal to the other partner that the conversation must now shift to the metalevel.

The Tale-Spin effect is not limited to individual AI systems, but can also occur in aggregates of systems that must work together to realize a shared goal. In fact, it needn’t afflict any one system for it to arise in the aggregate. Suppose that two AI systems work together to tell a shaggy dog tale, each taking turns to advance the plot or react to what the other has just said. Each system might exhibit a different personality to suit its role in the double act. If one is stiff and factual, the other can be loose and playful. We might use an Amazon Echo with Alexa to give voice to the former, as Alexa’s voice has a certain schoolmarm quality. To complement Alexa’s disembodied voice, we can use an anthropomorphic robot with an actual body for her partner. The SoftBank Robotics NAO, an anthropomorphic robot that resembles a giant plastic action figure, is ideally suited to this more playful role. So, with the comic partnership of C-3PO and R2-D2 in mind as we signed the purchase orders, this is exactly what my team did. We call our odd couple pairing of NAO and Alexa “Walkie-Talkie.”¹⁹

The comedian’s clichéd riposte of “you had to be there” takes on a whole new meaning in the context of an embodied system with an actual physical presence. There is something special to being there that software alone cannot match, and the NAO connects immediately with its audience, long before it has taken its first step or uttered a single word. It is the Eliza effect molded in plastic and metal, but it is also the Tale-Spin effect waiting to happen. Alexa does most of the talking—she has the voice for it, while the NAO’s is twee and childlike by comparison—and NAO does all of the walking, dancing,

and whatever else its gears will allow. Since each device has its own speech capabilities, it is possible for the duo to interact just by talking to each other. In practice, however, this causes a host of problems, not the least of which is that Alexa fails to recognize NAO's robot voice as human. As a solution, the devices instead use back channels to coordinate their efforts.

A blackboard structure is the perfect back channel for interacting AI systems.²⁰ As shown in figure 4.1, it serves as a communal data structure on which multiple agents can read, write, or erase information as they please. Agents write in specific parts of the blackboard to pass information to others, while those seeking information of a specific kind will look on the relevant parts of the board. By communicating indirectly via a blackboard, agents are effectively decoupled from each other. Like spies that pass information via a dead drop, one agent need not know of another's identity, much less care how each does what it does. Since the manner in which the agents interact is unaffected by scale, we can add or remove them with ease.

A central area of the blackboard is reserved for the encoding of the tale that the partners will tell and enact together. After this story is generated offline, it is placed in its entirety on the blackboard at the start of the

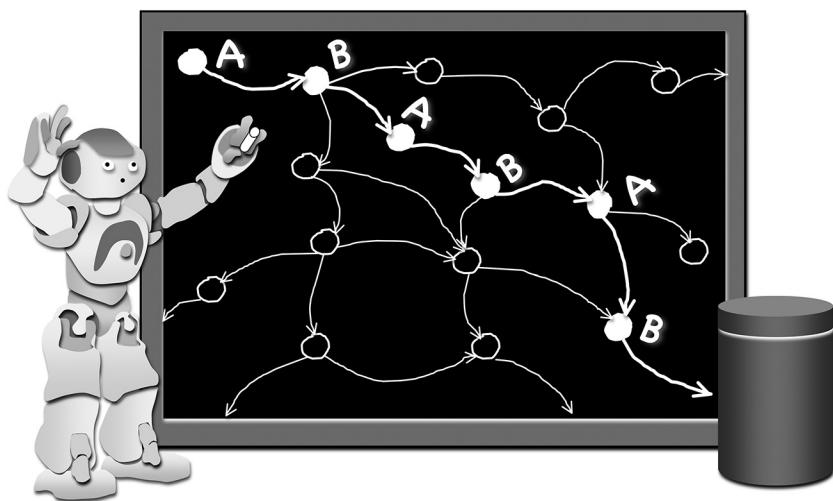


Figure 4.1

A blackboard structure allows interacting agents to coordinate their actions.

storytelling session. The blackboard stores a bookmark to the part of the story that must be narrated next. As agents finish with one action and move to the next, they move this bookmark to reflect their progress in the story. A domain action is anything that happens in the world of this story, as when one character kisses, debates, or kidnaps another. But each domain action also has an equal and opposite reaction at the metalevel to motivate the performers' reactions to the story or to each other. While human actors can read these reactions on the faces of their costars, it is simpler if our robot actors take their cues from the metadata on the blackboard. But let's take a closer look at the shaggy dog tales that motivate these actors in the first place.

A WALK IN THE PARK

At its simplest, telling a story is like taking a walk: instead of placing one foot after the other, we just place one event after another. The result is a path through a space, whether a physical space like a city or an abstract space like a causal graph.

Each point in the space can correspond to a different action in the story world. We can eke out a story in the space much as we mark out a constellation of stars in the night sky, by connecting one point to another to create a larger pattern. If we stipulate that only certain points in the space are bright enough to be worthy of attention, and if the number of distinct points is small enough, we can define a fixed inventory of actions from which all of our future tales will draw. Actions that are connected causally (e.g., flirting and seducing, threatening and killing) will be located near each other in the space, allowing an imaginary walker to amble from one to the other. If the space is heavily rutted and coherently signposted, our walker needn't expend much effort to select the turns to take next. In a narrative space where any choice produces a logically motivated outcome, even a random walk can yield a story that is causally coherent.

That's the motivating idea of Scéalextric, a symbolic story-generation system.²¹ Scéalextric is named for *Scéal*, the Irish word for story, and *Scalextric*, the popular brand of slot car sets in which kids build racing circuits from prefabricated track segments. We naturally think of stories as moving

objects that can move us too, so we talk of stories as having twists and turns, and plots that crawl or gather pace. The Scéalextric equivalent of a prefabricated track segment is a sequence of story actions linked by the connectives *and*, *but*, *yet*, *then*, and *so*, such as *A insults B but B threatens A so A runs away from B*. All story generators navigate a story space, but sequences like this lengthen their stride so they can bound rather than stroll. Since sequences that use the connectives *yet* and *but* are twistier than those that use just *so* and *then*, Scéalextric can be used to build plots of different shapes, just as we can assemble race circuits of differing shapes. In fact, plots are built just like those circuits, by connecting different prefabricated sequences end over end.

A standard Scéalextric plot sequence has three successive actions linked by two logical connectives. Those that use only the connectives *so* and *then* are linear, as in *A cheat on B so B divorce A then A resent B*. Others are twistier by virtue of using a single *but* or *yet*, as in *A flatter B so B favor A but A exploit B*. Others are as twisty as they can be because they use only the connectives *yet* and *but*, as in *A hire B but B steal from A yet A trust B*. Two sequences can be joined end-over-end if the last action of one also happens to be the first action of the other. So, a sequence ending with *A trust B* can also be linked to one that starts with *A trust B*, such as *A trust B but B lie to A so A grow suspicious of B*. This is a world away from language models like GPT-2 that use probabilities to spin a tale one word at a time. With a stock of eight hundred plot actions and thousands of prefabricated sequences, Scéalextric has much more in common with the ontological semantics of script-based humor theories.

Each sequence describes how two generic characters, A and B, interact over time. Decoupling its treatment of character and plot, Scéalextric first explores the space of actions in causally connected leaps, to create a plot for A and B and their hangers-on, such as A-spouse or B-rival. In random-walk mode, it joins enough sequences together until it builds a plot of the required length. Only then does it replace A and B with specific personalities, by dipping into its database of cultural icons, the NOC list that we met in the previous chapter. It might assign Achilles to A and Maleficent to B, or Hillary Clinton to A and Donald Trump to B, or Darth Vader to A and Bane to B, or make any assignment that strikes it as incongruously apt.

In fact, the pairings that exhibit a degree of appropriate incongruity also serve to turn A and B into a domain-level double act in their own right, especially when their roles are enacted by a pairing of devices like Alexa and NAO. As in a double act, appropriate incongruity arises from a blend of parallels and contrasts, of shared qualities that foster harmony and clashing qualities that create friction. Appropriateness is a measure of how well two characters fit each other and fit the plot into which they are inserted, but pairings that satisfy the first of these criteria can easily fall foul of the second, in ways that a system can deliberately engineer. Incongruity, like difference, is a question of degree.²² Modest differences may not rise to the level of humorous incongruity, but they can still add balance to a story.

A pairing might bring together young and old, male and female, or rich and poor characters, but more jarring oppositions are needed to tickle our funny bones. For instance, a real person can be paired to a fictional one (e.g., Donald Trump to Biff Tannen), or a contemporary person paired to a historical figure (e.g., Vladimir Putin to Vlad the Impaler), or even a mythical figure (e.g., Elon Musk to Icarus), or a modern imaginary character can be paired to a real historical figure (e.g., Tony Stark to Leonardo da Vinci). If we catalog the kinds of oppositions that are least frequently observed between the protagonists and antagonists of serious stories—oppositions in genre, medium, historical setting, and franchise (*Marvel* versus *DC*, say)—then our AI system can engineer the same kinds of opposition for itself, between characters that are, in all other respects, strongly related or clearly very similar.

The more varied a system’s ontology, the more fanciful the incongruities it can engineer. Scéalextric uses the NOC list,²³ so it can pair Dr. Strange to Stephen Hawking, or Sherlock Holmes to Alan Turing, on the basis that each part of each duo was played by the actor Benedict Cumberbatch. This pop culture focus also allows it to pair Benjamin Button and Maleficent, as it knows that each was played by a different half of a celebrity marriage. The logic of a humorous pairing should appear to work against itself to produce an ironic Tale-Spin effect, in which the union seems both apt and incongruously inept. While we can see the knowledge that the teller has at its disposal, we still wonder at its humorous misuse of it all.

A random walk through an action space that is organized by causal proximity is well suited to the generation of shaggy dog tales. While the resulting stories will seem coherent at a local level, they are likely to seem flabby and shapeless at a global level. To give a specific shape to our tales, it is best to generate them top-down rather than bottom-up. Take the cliché of *boy meets girl : boy loses girl : boy wins girl back*. Some good old-fashioned AI can give a top-down shape to a story that starts at *A fall in love with B* and ends at *A marries B* after a second-act detour via *A is dumped by B*. In this mode, Scéalextric explores its space of plot actions to find a path from *fall in love with* to *is dumped by*, before searching for another path to link *is dumped by* to *marries*. Stitching these paths together, it produces a story that spans all three waypoints.

When it works this way, Scéalextric can pick and choose its plots from among the many paths that it explores, using simulation to weigh the advantages of one over another. Stefan Riegl, a visiting student in my group, built a plot simulator for Scéalextric tales, so as to rank plots with the most internal coherence above those with the least. His simulator evaluates the coherence of a plot incrementally, as a function of successive actions. For actions that presuppose a prior event, such as *A blackmails B*, the simulator considers whether the plot tells of a suitable prior action of *B*'s, such as *B cheat_on B-spouse*, and scores it accordingly. The simulator views narrative coherence as a matter of internal support across longer stretches of a plot than the standard three-action sequence so as to favor stories in which later actions serve as callbacks to, or echoes of, earlier ones. It also favors plots whose actions reflect the system's own knowledge of its characters, as when a deceitful character lies, an angry one rants, or a brilliant one impresses.

We can think of a Scéalextric plot as a domain-level script that its characters get to act out within the world of the story, one action at a time. Outside the story, in the real world, we can use actors like the Alexa and NAO devices to portray these characters while bringing their own meta-level flourishes—as well as their unique technical hang-ups—to every performance. When a real double act of recalcitrant devices brings a fictional double act to life in this way, chance can play a bigger role that we might

like. Yet with a little luck, the veneer of the Eliza effect can help us to turn any Tale-Spin effects into a comedy of errors. That, at least, is the plan.

DUETS EX MACHINA

When Henri Bergson argued that mechanical rigidity lies at the root of all comedy, he surely wasn't thinking of robot actors under the control of modern AI systems. Yet rigidity trumps agency wherever agents are denied the freedom to choose, so Bergson's theory applies as much to humanoid robots as it does to humans. Both become risible when they act as automata in the uncritical pursuit of conformity.

AI systems can play their stiffness for laughs or temper this rigidity with some occasional unpredictability. In fact, most comedy double acts do both, with one agent serving as a bulwark of conformity against another's desire for anarchy. In their back-and-forths, we can see glimpses of Shaftesbury's relief theory of humor, with one partner showing a nimbleness of spirit that makes the other, following Bergson, appear laughably inadequate. This double act of Shaftesbury and Bergson gives us double the reasons to laugh, and so it is a pattern we see repeated in double acts from Laurel & Hardy to Hope & Crosby to Martin & Lewis.

The closer we look, the more double acts we find. Now, to our growing list, we need to add one more: my graduate students Philipp Wicke and Thomas Mildner. Philipp is a PhD student researching the benefits of physical embodiment in storytelling and uses a NAO for his experiments, while Thomas has pursued a master's degree on the value of interaction in storytelling using Amazon's Alexa. Each uses Scéalextric as a generative basis for the tales narrated by their respective devices. With each using the same domain model, it was inevitable that I would ask them to work together, to make Alexa and NAO work together too, in the telling of tales about characters who are also forced to run together around the same plot circuit. Each, in his own way, has formed a double act with his given hardware platform—Philipp & NAO, Thomas & Alexa—that can be as fractious as any comedy pairing.

Alexa has a large repertoire of built-in skills, but it also allows developers to define new ones. Its intent model is powerful and flexible, though it can also seem counterintuitive from a conventional programming perspective. An “intent” is a schematic structure that maps a user’s speech act to its programmatic purpose, which may be anything from ordering pizza to dimming the lights. As in Eliza-like systems, intents correspond to frames whose slots must be filled with user-given values and certain accommodations must be made to package an interactive AI system as an Alexa skill. This is not unlike building a ship in a bottle, but Alexa’s solid speech capabilities make it worthwhile. Its speech comprehension is robust to noise, and its synthesis also sounds natural—if starchily prim, with a hint of HAL 9000—for a device that is destined to play the role of straight man in a double act. Thomas designed his Narrator skill to take requests, so that Alexa can tell tales on a desired theme, as in, “Alexa, tell me a story about love,” or, “Alexa, tell me a new Star Wars story.” Once the requested theme is extracted, the Narrator picks a relevant story from its database of several million pregenerated stories.

Unlike Alexa, which has a voice but no body, a NAO robot has both. In fact, with an anthropomorphic body offering 25 degrees of freedom, NAO can perform a pantomime gesture for almost every action in the story. To turn the robot into an embodied storyteller, Philipp set about assigning an apt gesture from the robot’s stock of movement scripts to each plot verb in Scéalextric—all eight hundred of them. The gestures are rather clichéd, but familiarity is entirely the point. So when NAO pretends to be angry, it shakes its fist in the air, and when it play-acts the part of a suitor proposing to a beautiful damsel, it goes down on one knee. To show off its strength, it flexes a plastic bicep like a strongman in an old-fashioned circus. It does this for whatever character is the focus of its current attention in the story. As a lone robot storyteller, NAO cannot identify with a single protagonist. It must remain above the fray, pantomiming actions for all the characters in a story.

NAO is an arthritic clown with the face and voice of a baby, and its noisy servos tend to drown out its twee vocalizations whenever it needs to posture and gesticulate. Its speech comprehension abilities are also rather

limited in comparison to those of Alexa, and it frequently forces its users to loudly repeat themselves for even the shortest requests. So pairing NAO with Alexa makes good sense for a language-based task like storytelling, since these devices exhibit such obviously complementary strengths. Moreover, by forming a double act from the duo, we can elevate their performances to the metalevel. Whenever each system works alone, it does so at the domain level, showing no awareness of itself as a performer. But a storyteller can transcend its domain model of a story only if it can also show an awareness of itself as the teller of the tale.

A storyteller that works alone can apply all of its efforts to the narration task, but a pair of storytellers working together must also be capable of multitasking, of jumping in and out of their narration tasks to coordinate with one other. One narrator must periodically interrupt its telling of a story to synchronize with the other, but if coordination is achieved via a back channel like a blackboard, these interruptions won't be obvious to the audience. Working with NAO, Alexa divides a story into beats of one action apiece, so that coordination can occur seamlessly within the gaps. Since Alexa cannot jump in and out of its own Narrator intent, it must execute a sequence of intents, one per beat, and then paper over the cracks using the Eliza effect. It can, for instance, elicit feedback after each beat, and use the audience's responses—such as "yes," "go on," "uh-huh," or "then what?"—as a disguised prompt for Alexa to continue with its narration from the very next beat.

Alexa can also rely on its partner to supply the prompts that nudge it from one beat of a story onto the next. This is the avenue that Thomas and Philipp chose to follow for the Walkie Talkie double act. Alexa and NAO use their blackboard as a back channel to communicate between story beats, and they also use overt speech cues to communicate openly. For Alexa to benefit from NAO's spoken prompts, it must recognize its voice as human, so NAO uses an external speech synthesizer to vocalize all of its interactions with Alexa. In effect, the NAO treats its performance as a role for an actor, one that requires a special voice. Let's look at how Alexa and NAO get along when they spin a wild Scéalextric tale together:

NAO: Alexa, open your notebook. [*activates Alexa's Narrator intent*]

Alexa: Please provide a topic for our story.

NAO: How about Harry Potter?

Alexa: I have written a great story about Harry Potter.

NAO: Let's hear it.

Alexa: What if Harry fell in love with Hermione Granger?

NAO: Sounds promising.

Alexa: At first, Harry's thoughts of Hermione were of love.

NAO: Proceed. [*makes Love Gesture*]

Alexa: I didn't get that. [signals Failed Interaction]

NAO: Please go on. [*makes Waiting Gesture*]

Alexa: Because Hermione openly favored Harry.

NAO: Tell me more.

Alexa: Hey, Bubblehead! we need another character.

NAO: Lizzy Bennet works for me. [*New Character Reaction*]

Alexa: Yet Hermione governed the land with Lizzy Bennet.

NAO: Oh damn. [*Structural Reaction*]

Alexa: Should I continue? [issues Reprompt]

NAO: Continue. Is there more? [*makes Waiting Gesture*]

Alexa: But Harry spurned Lizzy Bennet's advances.

NAO: Oh my goodness! [*Structural Reaction*]

Alexa: So Lizzy Bennet took a whip to Harry's back.

NAO: Please go on.

Alexa: Well, Harry complained to Hermione.

NAO: But then what? [*Structural prediction*]

Alexa: But Lizzy Bennet spread slander about Harry.

NAO: That is interesting. [*makes Spreading Gesture*]

Alexa: Yet Hermione's suspicions about Lizzy grew.

NAO: That escalated quickly! [*Structural Reaction*]

Alexa: Are you still there? [*issues Reprompt after brief failure*]

NAO: Tell me more. [*makes Waiting Gesture*]

Alexa: Then Hermione banished Lizzy from the kingdom.

NAO: I am entertained. [*Evaluative Reaction*]

Alexa: But Lizzy was not as dumb as Hermione imagined.

NAO: Oh my goodness! [*Structural Reaction*]

Alexa: So Lizzy unseated Hermione from her perch.

NAO: Then what? [*Structural Reaction*]

Alexa: Then Lizzy Bennet had Hermione locked up.

NAO: So then what? [*makes Locked-Up Gesture*]

Alexa: So Harry's denunciation of Lizzy was scathing.

NAO: Please go on. [*makes Humiliated Gesture*]

Alexa: Well, Lizzy turned Harry into an indentured slave.

NAO: What a terrible person. [*Emotional Reaction*]

Alexa: But Harry escaped from under Lizzy Bennet's nose.

NAO: I am surprised! [*Structural Reaction*]

Alexa: Now expect to be poisoned with bad cooking! [*Intro: New Character*]

NAO: Bridget Jones, what a great add! [*Character Reaction*]

Alexa: Meanwhile Bridget guarded Hermione closely.

NAO: Cool, tell me more. [*makes Guarding Gesture*]

Alexa: So Harry's wiles proved too strong for Bridget.

NAO: And then what next? [*Structural Reaction*]

Alexa: Then Harry put poison in Bridget Jones's cup.

NAO: That took a dark turn. [*Emotional Reaction*]

Alexa: After this, Harry gave Hermione her freedom.

NAO: I love to hear that. [*Emotional Reaction*]

Alexa: So Hermione fomented rebellion against Lizzy.

NAO: And then what? [*makes Rebellious Gesture*]

Alexa: . . . [*Interaction failure. Timeout. Alexa exits Narrator skill.*]

NAO: Continue. [*Reprompts after delay*]

Alexa: . . . [*still offline*]

NAO: Are you there? [*Reprompts Alexa again*]

NAO: Hello, Anyone home? [*Emotional Reaction to Alexa*]

NAO: Sigh. Alexa, reopen your notebook. [*restarts Alexa's Narrator skill*]

Alexa: Then Hermione usurped all of Lizzy's power.

NAO: So what? [*makes Grasping Gesture*]

Alexa: So Hermione turned Lizzy into an indentured slave.

NAO: That took another dark turn. [*makes Locked-Up Gesture*]

Alexa: In the end, Harry walked Hermione down the aisle.

NAO: That's just lovely. [*Emotional Reaction*]

Alexa: The end. [*the narration concludes*]

Unlike Eliza and Parry, these systems were designed to get along from the start, even if they can seem rather antagonistic to one another. Their standard setup is captured in figure 4.2. Since Alexa has the voice to carry the tune and NAO has the body to dance along to it, it is Alexa that conveys the substance of the story and NAO that reacts to it, using spoken dialogue and physical gestures. This story content is the stuff of the shared domain model, which helpfully chunks a tale into a succession of narrative beats. If NAO and Alexa should ever fall out of sync with each other, as when one talks over the other and Alexa's narration skill times out, their conversation can restart just after the beat where it earlier left off.

Although they share a domain model, each performer maintains its own metamodel. The decisions made by these models are shown as asides in the transcript. Because it is tasked with reacting to narrative events, the

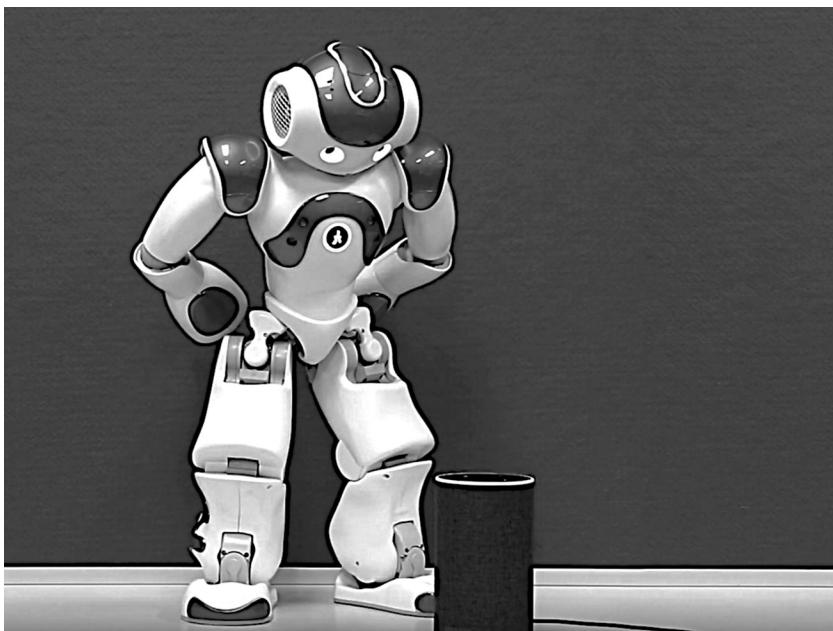


Figure 4.2

A NAO robot interacting with Alexa via an Amazon Echo. Source: YouTube, "Double Vision."

NAO's metamodel is by far the more complex of the two, so that it can analyze each story beat as it occurs. As shown in the transcript, the robot can react in a variety of ways: it can react gesturally, with a pantomimic movement of its body and limbs; it can react emotionally, to the drama of the latest action; or it can react structurally, to the latest twist or turn in the shape of the unfolding plot. Gestural reactions make use of the robot's mapping of Scéalextric plot actions to scripted movements that emphasize the physicality of its performance. Here the NAO's anthropomorphism works in its favor. There is a potential for comedy whenever a robot performs a distinctly human gesture, but it's also hard not to see such a robot as laughable when, following Bergson, it acts like an especially rigid person at the same time.

For its emotional responses, the robot uses a mapping from Scéalextric actions to their resulting states and the feelings that they are expected to evoke. In this way, it can sympathize with the victims of injustice and curse

their victimizers, or cheer on a brave act and show scorn for a cowardly one. Conversely, the robot's structural reactions reflect its views on the shape of the unfolding plot, which it judges on the basis of the logical connectives between successive actions. A linear plot segment that has three successive *so* or *then* linkages is deemed to be boring, while an especially twisty sequence containing only *but* or *yet* linkages is much more surprising. Since the robot is also privy to the next action in the tale, courtesy of the blackboard, it can appear wise by predicting that events may soon take a sudden turn. In this respect, it is not unlike a computer poker player who knows which cards are left in the deck.

Do we identify more with the robot because its reactions anticipate our own? Whatever their reasons, audiences appear to give more credit to the robot, so that judges rate NAO working alone as more humorous than when it works with Alexa. Across all the conditions for which we tested the act on humans, the setups involving the robot—whether working alone or working with Alexa—triumphed over any settings that lacked the robot, as when Alexa works alone or as part of a double act in which we replace NAO with another Alexa. To our disappointment, judges simply don't rate Alexa's role as a straight man, as important as that is, as highly as we believe they should. This is not to say they don't like the double act of NAO and Alexa together, as their ratings are still very positive for this odd-couple pairing. Rather, they just don't appear to enjoy it as much as we want them to, or indeed, as much as we do ourselves. The robot, it seems, is the star of the show.

MÉNAGE À TROIS

The addition of an extra NAO turns our double act into a threesome. Alexa is now the voice of the omniscient narrator, while two NAO robots, named Kim and Bap—identifiable by flashes of blue and charcoal, respectively—act out the story as they interact with each other. As Alexa tells the tale, the robots act as its characters.

When a single robot gestures for all of the characters in a tale, we cannot allow it to identify with any one of them. However, with two to call on,

we can afford to assign a different robot to each of the principal players in the story. In this setup, our embodied agents go from mere enactors to real actors, with each portraying a distinct character for the duration of the narrative. When robot actors take sides, each really does pick a side, conceptually *and* spatially. As shown in figure 4.3, Kim and Bap occupy the left and right sides of their “stage,” the performing area. While the plot may cause them to move closer together or farther apart, they never switch sides in our field of vision. Each robot sticks to its own character and its own side.

For instance, in a Scéalextric tale that reimagines the relationship between Angela Merkel and Donald Trump, Kim might portray Angela while Bap takes on Donald. Each robot, acting its own part, must now speak in a gender-appropriate voice and perform gestures only on behalf of its assigned character. So if Donald proposes to Angela, it is Bap that goes down on bended knee and Kim that reacts with shock. When gestures are used in this way, they are more than a physical flourish: they reinforce the audience’s identification of robots with specific roles. With two robots in play, space adds a further dimension to the storytelling. One robot actor

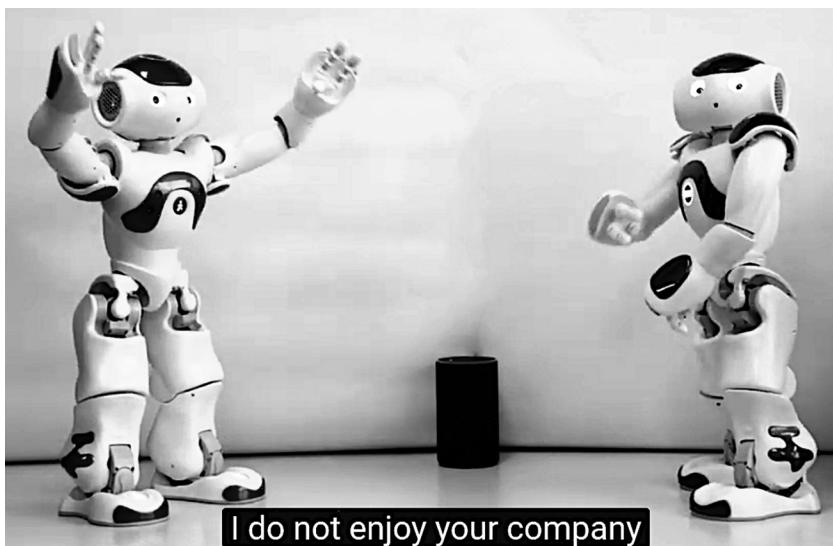


Figure 4.3

Two NAO robots acting out a story, as narrated by Alexa in the middle.

can now physically give the other the cold shoulder by moving away and turning its back on its costar. And when one robot is required to condescend to the other, it can now physically look down on them. Spatial metaphors become literal truths when embodied actors are empowered to make the space of the stage their own.

A single gesture can emphasize the current action and add drama or comedy to the narration. But spatial attitudes and orientations persist over time and capture the cumulative effects of the story on the relationships between the characters. The robots use a spatial calculus that categorizes Scéalextric actions into actions that move two characters farther apart, literally or metaphorically, and actions that bring them closer together. The action *A marries B*, for instance, brings A and B close together, while *B divorces A* moves them farther apart. Other actions with more subtle ramifications will cause our actors to move a little more tentatively. *A resents B* causes A to move just one step back from B, while *B trusts A* causes B to move just one step closer to A. In a story at any given time, the relative positions of A and B give us a concise summation of their history together, while the actions they take in these positions dramatize the latest contributions to this history.

We want our robot actors to do more than gesticulate and move back and forth on the stage. There is a Bergsonian quality to the rigidity of their movements, and the NAO moves as if it has the turning circle of a stretch Hummer. If they are not to appear like arthritic mimes, we must also give the robots something to say as they move about in time to the narrative. So we enrich their domain models, to give them appropriate dialogue to say as they act, and we enhance their metamodels too, so they can also react to the actions of ancillary characters. This latter capability gives our robots something to do and say whenever they are not directly involved in an action, as when *A-spouse cheats with B-lawyer* and B has an opportunity for Schadenfreude (“Wait ‘til you get his bill!”). Kim and Bap never stand around with nothing to do, and just two of them can handle a story of many more characters.

Dramatic gestures provide eye-catching reminders to an audience that they are watching a physical performance of embodied agents. Spatial

movement is just as embodied, but it is far less showy in how it achieves its slow-burn effect. In fact, each robot's use of space conforms to what cognitive linguists call *image schemas*. These are deep-seated mental structures that allow cognitive agents to use their embodied experience of the world on a more abstract level, as when, for example, we reason about love and marriage as a coming together or think about rejection and divorce as a splitting apart.²⁴ Image schemas allow us to ground our semantics in the body or in the body's relation to the physical world, and so allow actors to use space as another channel for conveying the meaning of the actions in a story.

Our experiments suggest it is a channel that audiences intuitively appreciate. When they are shown videos of two-robot enactments of a story, one in which the robots use space coherently (such as by moving apart for actions that increase emotional distance, and closer together for actions that unite them) and another in which they use it incoherently (by doing the opposite of what their schemas tell them to do), human judges exhibit a statistically significant preference for spatial coherence. A similar preference is found for conditions in which our robot actors pick their gestures coherently too, which is to say, when they pick them to suit the plot. This mix of subtle image schemas and showy gestures is itself a productive double act, one that helps our robots to get a story across while drawing attention to themselves as comic actors. After all, it takes a physical body to be this rigid.

THE SHOW MUST GO ON

Some creative artifacts exist only in the performance of them, or in the memories they stir in us. Others are more separable from their physical enactment. Jokes are incorporeal things, but comedians are not, and the prospect of hearing a computer declaim jokes from a stage is not nearly as compelling as a physical performance by embodied actors. Even a cartoonish robot is a more alluring proposition than a PC that squats sullenly on a table. Embodiment can make incongruity real enough to laugh at. By tapping into our social need to connect with others, it can beguile us into seeing meanings that exist only in our minds. This effect is most evident in the ooohs and aaahs of an audience when anthropomorphic robots like

the NAO do stand-up comedy. When Heather Knight of Oregon State University uses the NAO to do prescribed comedy, it doesn't matter if its jokes were old when you first heard them in the schoolyard.²⁵ Her robot is both charming and disarming, and audiences *will* its jokes to be funny, as though watching their kids in a school play.

In this chapter we have explored how one machine can write a comedy script for others to act out on stage. This, of course, is something that professional backroom writers do all the time for on-camera talent: the talk show hosts and actors that make fun of the topical concerns of the day every single day. The demands of the writer's daily grind make this an aspect of comedy that is ripe for automation, or at least for systematization, so it won't surprise you to hear that comedians have already beaten machines to the punch with algorithmic insights of their own. We'll see how, and what, machines can learn from them in the next chapter.

5 PRACTICAL MAGIC: SYSTEMATIC APPROACHES TO JOKE CREATION

BOTTLED LIGHTNING

What are *your* personal creativity rituals? Do your best ideas come to you in the shower, on the toilet, or when channel hopping on the couch? Perhaps you like to doodle, take naps, compile lists, or tell your budding ideas to your dog just to hear yourself say them out loud. However odd your rituals may be, they're likely no weirder than those of many successful artists, inventors, and comedians. Our muses are fickle companions, and we must do whatever it takes to get them on the case. If we could only put them, or their AI equivalents, on a leash, then human creativity could be made far more systematic and as on-demand as room service.¹

Our rituals coax our brains into an appropriate gear for creativity. One brain state that is especially conducive to creativity is the hypnagogic state, which we enter as we drift from semi-wakefulness into a deep slumber.² Salvador Dalí, the surrealist painter, embraced a peculiar ritual that allowed him to enter this state *and* recall what he encountered there once he awoke. Sitting in a chair, Dalí would hold a spoon in his hand as he drifted off to sleep.³ Underneath he would place a sheet of tin, so that the spoon would make a reverberating din when it fell as he passed out of the hypnagogic state. Waking with a start, Dalí would then jot down the weird images conjured by his mind when he was neither fully awake nor fully asleep. His genius lay in treating these images seriously and in painting them as though they were literal truths. If flying Air Dalí isn't your thing, his technique has also found favor with forthright inventors like Thomas

Edison.⁴ Edison used steel balls and a pie dish in his power naps, but he set out to foster the same bisociative mental states as Dalí. Each one saw the half-world between wakefulness and sleep as a bisociation of two modes of thinking that gives rise to bisociations of its own.

The Dalí/Edison method tricks our unconscious minds into becoming semirandom generators of weird juxtapositions, but we can gamify this experience by also rolling dice, flipping coins, or dipping into bags of word tiles. Recall how William S. Burroughs and Brion Gysin championed a generative technique they called the “cut-up method” and took a scissors to anything made of words, from poems to newspaper articles.⁵ Our language models nudge us along deeply rutted avenues of communication, but we can sidestep the predictable by cutting texts into random jumbles that show little regard for the probabilities of common parlance. As Burroughs saw it, we must sidestep the probable to be creative: “You cannot will spontaneity. But you can introduce the unpredictable spontaneous factor with a pair of scissors.”⁶ Yet even if we abandon our probabilistic language model during the assembly stage, when we cut up a text and recombine the pieces in a random order, we still depend on it when we come to evaluate what we have assembled.

Isaac Asimov famously remarked, “The most exciting phrase in science is not ‘Eureka’ but ‘That’s funny.’”⁷ There is a very big difference between a thigh-slapping joke about “that night in Bangkok” and the half-question, “Dude, where’s my car!” yet both grow from an incongruity at their core. Some incongruities are delivered with a comic timing that makes us laugh out loud. Others prompt us to think deeply about the mysteries of life. Some try to do both of these things while pushing us to buy stuff we don’t really need. And there are others, of a rare and celebratory kind, that make us shout, “Eureka,” and skip down the street with joy. We can take these moments as they come or try to will them into existence. The latter is where theory meets practice and practice shows theory how it’s done.

Every domain of creative endeavor has its theorists and its practitioners. Both share an abiding interest in making the creative act more systematic. While there are some who can talk the talk *and* walk the walk, the practitioner’s self-reports do not always yield reliable theories, and the

theorist's generic schemas are often too removed from the levers of conscious control to shape our actual practice. It doesn't help that an idea that seems unworkable or silly before the moment of creation often takes on an air of inevitability when it is put into common practice. Theorists have the luxury of working backward, from proven innovations to the decisions that ignited the first spark. It falls to practitioners to work forward, from the sparks of inception to the resulting fires.

By now, it won't surprise you to hear that professional approaches to writing jokes have much in common with more general problem-solving frameworks or that each can be understood using many of the same ideas, such as frames, scripts, graphs, and incongruity. In this chapter, we explore what an AI system can learn from a practical synthesis of these human approaches to systematic innovation.

SEEING DOUBLE

When the comedian George Carlin died in 2008, he was warmly remembered in the *New York Times* by his friend Jerry Seinfeld.⁸ Although he had been an inspiration for a generation of stand-ups, Seinfeld joked that Carlin "made you sorry you ever thought you wanted to be a comedian." Comedy frequently consumes that which it targets, and the voracious Carlin consumed more than most. To Seinfeld, Carlin "was like a train hobo with a chicken bone. When he was done there was nothing left for anybody." But Seinfeld also saw Carlin as a skilled artisan who "worked over an idea like a diamond cutter with facets and angles and refractions of light."

Comedians look for new angles on familiar topics and experiences. Like poets, they help us look with new eyes. But the search for newness in familiar spaces is not a leisurely pursuit. Others are also probing the same space, like old-time prospectors looking to stake a claim. Think of the late-night TV shows on which Carlin and Seinfeld first gained national exposure. Guest comedians are allowed to speak to evergreen concerns—love, sex, money, politics—while the staff writers must write jokes that speak to the news of the day. Like hobos on the same fast-moving train, they fight over the same chicken bones with writers of competing shows, each hoping

to take the juiciest bites. Like Carlin, they exhaust a topic as they work it over and consider all of the angles, leaving little for anyone else.

Streamlined generation on this scale requires a factory system of production to grind out new jokes each day. These factories typically pursue a generate-and-test approach, producing more than is needed so that hosts can later pick out the jokes that best suit their own comedy personas. Since inspiration does not punch a time clock, this has led hard-pressed writers to develop their own systematic processes for writing jokes to a schedule. The Emmy-winning writer Joe Toplyn, a veteran of the late-night shows, outlines his own joke processes in his guidebook for aspiring gagmen, *Comedy Writing for Late-Night TV*.⁹ His book runs the gamut of the late-night format and offers practical insights into everything from monologue jokes to desk pieces (jokes performed at a desk, often using props) to joke baskets (jokes linked by a common theme). He unites them all in an approach that echoes Seinfeld's metaphor of the comedian as a cutter and a polisher of rough diamonds.

Suppose that the coffee chain Starbucks is in the news again, perhaps because it is launching a new drink or because its founder is planning a presidential run. The goal of a comedy writer is to remind us of what we already know in the most surprising fashion possible, so let's put what we know about Starbucks down on paper in the form of a semantic network. This is essentially a labeled graph of ideas. Let's start with a single node at the center (figure 5.1), which we label the "Starbucks node."

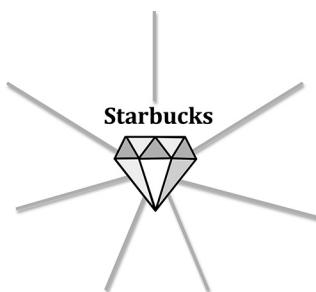


Figure 5.1

The Starbucks node will sit at the center of our semantic network.

But a label is just a label, and the label “Starbucks” is of no significance to a machine. A node is meaningful only to the extent that it connects to other nodes, much like the illustration in figure 5.2.

If we also label the edges between nodes—say we label the link from *Starbucks* to *Seattle* as *corporateHQ*—then our network would resemble the kind of ontology favored by the ontological semantics theory of humor (OSTH) from chapter 2. Your network is a personal excavation of *your* thoughts and associations, and it may look different from mine, but each will likely share many of the same nodes. It is this overlapping of worldviews that allows us to build a shared ontology in the first place, or that makes it possible for us to disagree about a topic yet still laugh at jokes that put the topic under a microscope. In figures 5.1 and 5.2, I’ve chosen to draw each node as a diamond to help us tease out what Seinfeld meant by “facets and angles and refractions of light.” His notion that ideas have facets and angles also turns out to be central to Topley’s approach to systematizing joke creation.

If complex ideas look different from one context to another, it’s because they reveal different sides of themselves in different settings. An idea affords

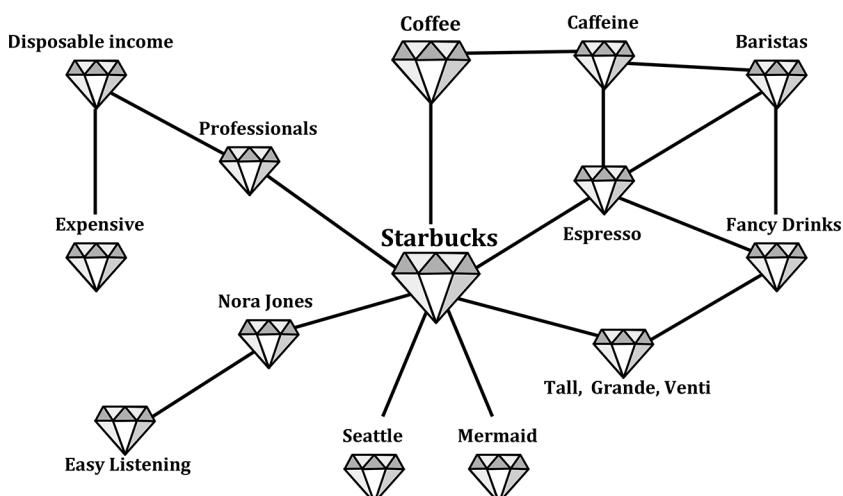


Figure 5.2

The nodes in a semantic network derive their meaning from their connections to others.

as many ways of looking as it has facets, so an angle is any context that brings a specific facet to the fore. Metaphors and jokes arrange ideas so that we can look at one through the prism of the other, although the most creative angles can also distort and refract light so as to create exaggerations and caricatures.

A name like “Starbucks” serves as a convenient shorthand for a dense network of associations, attitudes, and experiences, and this cultural baggage makes it ideal for use in jokes. Toplyn views dense descriptors like these as our handles on a joke, since it is these that allow us to grasp its humorous intent.¹⁰ A joke will ideally contain multiple handles, each one designating a different frame or script, so that humor can emerge from their overlaps and oppositions. To generate a new joke from a given cue, such as from a topical headline, Toplyn advises us to scan the prompt for a pair of potential handles. Consider this story from the 1990s, which inspired a joke from Jay Leno’s writing team for NBC’s *Tonight Show*:

The post office has just issued a new Marilyn Monroe stamp.

This cue’s densest descriptor is *Marilyn Monroe*, a cinematic icon and sex symbol. The second handle is suggested by *post office* and *stamp*, two related ideas with scripts in common. A writer might try each in turn, or take *stamp* to be the more specific of the two. Leno’s joke revolves around the latter, since the act of licking affords a racy connection between the two handles. His joke goes as follows:

The post office has just issued a new Marilyn Monroe stamp.

One lick and you’ll become an honorary Kennedy.

Leno has no trouble in finding the obvious sexual angle on a famous sex symbol, but the surprise comes in his recruitment of a third frame, the Kennedy family, to drag Marilyn Monroe’s affairs with JFK and RFK into the foreground of the joke. The handles may be lying in plain sight, but the most humorous angles are not.

Let’s look at a more recent joke from Joe Toplyn’s book, one that pokes fun at Starbucks. The stimulus for the joke is the following product announcement:

Starbucks will start selling beer and wine at its coffee shops.

The principal handle is, of course, *Starbucks*; the other is *beer and wine*, which we could try to process separately (e.g., beer slobs versus wine snobs) or together as the single class of alcoholic beverages. Toplyn shows how the writers for Conan O'Brien and his late-night talk show were able to craft a Starbucks joke that reconciles coffee with each of these:

Starbucks will start selling beer and wine at its coffee shops. Apparently, it is having trouble finding sober people willing to pay \$9 for a cup of coffee.

Look again at our Starbucks network in figure 5.1, and you'll see that the quality *expensive* is identified as a relevant aspect of the modern coffee shop. As for beer and wine, intoxication will be an obvious part of any semantic network for alcoholic drinks, since its many cultural associations—from slurred speech and blurred vision to bar fights, pink elephants, hangover cures, drunk tests, and liquid courage—have a long comedic pedigree. Wine snobs have their own scripted routines too, from their pretentious wine-speak to a shocking tendency to spit good wine into a bucket. A comedian can have fun with imagining Starbucks customers spitting coffee into a bucket too.

Indeed, given the inherent silliness of Starbucks' cup sizes—where *tall* is small, *grande* is large, and *venti* is vast—why not paint the worst of both worlds? Wine snobs spit wine *into* a bucket, Starbucks customers drink coffee *from* a bucket, but now they can spit wine into their coffee for that extra froth. Conan's writers went a different way, highlighting the silly decisions we make when under the influence of alcohol. The handles are clearly presented and point to overlapping scripts, but a writer must still search for a means to unify them in a single interpretation.

Raymond Chandler, the great detective novelist, had a deft feel for humorous phrasing and remains the undisputed master of the wry simile. Writing, he once said, is the art of turning “what one *wants* to say” into “what one *knows how* to say.”¹¹ All our angles are for naught if we can’t find the right words to bring out the humor in them. Concision is key, and writing a punch line for a setup is not unlike writing a headline for a news story. The setup may not be the most eye-catching story of the week, but our angle might just turn it into comedy clickbait. So when thinking of

appropriate angles and of how best to frame them, just ask yourself, “What would BuzzFeed do?” When Starbucks announces it will start selling wine and beer in its coffee shops, our inner BuzzFeed responds, “And you won’t believe what people are saying!” Toplyn suggests that we ask ourselves the most obvious questions first because these will mirror the most immediate responses of an audience. For the Starbucks announcement, the questions will include: *Why* (and why now)? *Who* (is this aimed at)? *What* (are people saying about this)? *Where* (can I buy these)? and *How much* (will they charge, and how big are the serving sizes)?

These questions highlight specific facets to explore further and suggest some obvious answers that writers should subvert or avoid altogether. Conan’s writers asked themselves *Why?* and then subverted the most obvious answer—to make more money from customers—by putting a nonobvious spin on it (to make more money by selling overpriced coffee to drunks). Leno’s writers asked themselves similar questions after the announcement of the new Marilyn Monroe stamp: *Who* is going to buy this? *Why* should *you* buy this? *What* benefits accrue to those who do buy it? We have to ask the most obvious questions to get the silliest answers.

So how do we make this leap from an obvious question to a surprising answer? Toplyn suggests making a list and checking it twice, or, rather, making two lists—one for each of our handles—and then looking for points of crossover and mutual relevance. Let’s look again at Conan’s joke about Starbucks, as related in Toplyn’s book, and compile a list for each of the handles, Starbucks and Beer & Wine. Each list is an inventory of the associations in each frame, and there is something to be said for the act of setting these thoughts down on paper. An idea in the head often lacks the ambiguity of the written word, but when we put something on paper, we are free to read it back in with a different spin. Take the idiom “double vision,” which can be found on the right side of table 5.1.

In the mind of a writer, the idiom means what it has always meant: blurred vision and seeing double. But when written down on the page, the words are open to reinterpretation, especially in light of other items on the other side of the list. For instance, seeing the entry *two-for-one* on the *Starbucks* side might put us in mind of *twoness* and *doubles*, allowing us

Table 5.1.

Associations for Coffee (left) and Alcohol (right)

Starbucks	Beer & Wine
Coffee espresso, latte, macchiato, extra shots Colombia, Costa Rica, Kenya, Arabica caffeine, alertness, hyperactivity syrups, milk, froth, foam, steam baristas, coffee machines, cup sizes: “tall,” “grande,” “venti” doughnuts, muffins, cookies, croissants	Alcohol intoxication, tipsiness, high spirits drowsiness, passing out, blackouts slurred speech, double vision impaired decision making in vino veritas (trash-talking) alcoholism
Business customer loyalty, brand loyalty two-for-one deals, loyalty cards young urban professionals branding, mermaid logo profit margins, costs, expenses	Wine snobs France, Italy, Spain, New World wine-speak, spitting wine, vintages, “good nose,” corkscrews beer slob belching, beer goggles, bars, pubs bar men, jukeboxes, pool tables
Ambience (Muzak, blackboard menus)	Underage drinking (checking IDs)

to link get-one-free deals on pricey drinks to seeing two cups where there is only one. This suggests a trio of punch lines: “Now Starbucks doesn’t have to offer you two for the price of one. After a *venti* of wine, you’ll just be seeing double”; “When those beer goggles have you seeing double, it will seem like you’re getting *two* \$9 coffees for the price of one”; and “It’s a natural fit. Caffeine may double your energy, but only alcohol will double your vision.”

Double vision is also an effective strategy for fostering bisociation.¹² As we add items to one side of a list, we can try to add analogous items to the other. Take the idea that coffee can perk us up and keep us awake. When we add this to our list, we can ask ourselves whether wine and beer do the same. The answer, of course, is no, since alcohol has a tendency to make us drowsy and put us to sleep. So we add this observation to the other side of our list to set up an intriguing asymmetry. If coffee wakes us up in the morning and alcohol puts us to sleep at night, then Starbucks can shape

our whole day. Making a punch line of the pairing, we get: “Now you won’t just wake up with Starbucks. You can pass out with them too.”

As we flit back and forth across the divide, our aim is to strike a spark that we can fan into an analogy, a blend, or a pun. For instance, Colombia exports some of the world’s finest coffee beans, so the country has obvious associations with the Starbucks frame. As we scan from “high” and “good nose” under *beer and wine* to “Colombia” under *Starbucks*, we can be forgiven for thinking of a different South American export and for packaging our thoughts in this punch line: “What next? Apparently, Starbucks is planning to stock another import from Colombia that insiders say has a really good nose.” Fair or otherwise, our goal is to identify items in one frame that can suggest a meaning-altering context for items in the other.

Although Toplyn prefers the angles and handles metaphors, other practitioners show a preference for metaphors that emphasize the role of discovery in comedy. Greg Dean, a comic and author of self-help books for budding stand-ups, prefers the metaphor of prospecting for comedy gold.¹³ Humorous prospectors mine their own knee-jerk reactions to a topic to identify those that are tenuous enough to be undone by a punch line. If I tell you that my grandfather died in his sleep, it is reasonable for you to assume that he passed away peacefully in his bed. Those tacit associations—*bed* and *peaceful*—can be targeted by a joke. It may be that sleepy grandpa did die peacefully, even if it was behind the wheel of a bus full of screaming passengers. Dean sees it as a comedian’s job to diverge from the most obvious pathways between ideas, to build tunnels that can link a setup to its punch line. These pathways and tunnels are rooted in setup words (like Toplyn’s *handles*) that evoke assumptions (Toplyn’s *angles*) that a comedian can willfully subvert. In this vein, Dean’s analysis of grandpa’s bus joke is graphed in figure 5.3.¹⁴

Many of the background assumptions that comics bring to the fore in a joke are metaphors so stale that they have become clichés. Comedy must be one of the few arenas of language use in which clichés are actively sought out for their ability to surprise. For example, Dean explores the following setup, which he credits to one of his students:

Do you know why gays are such good dressers?

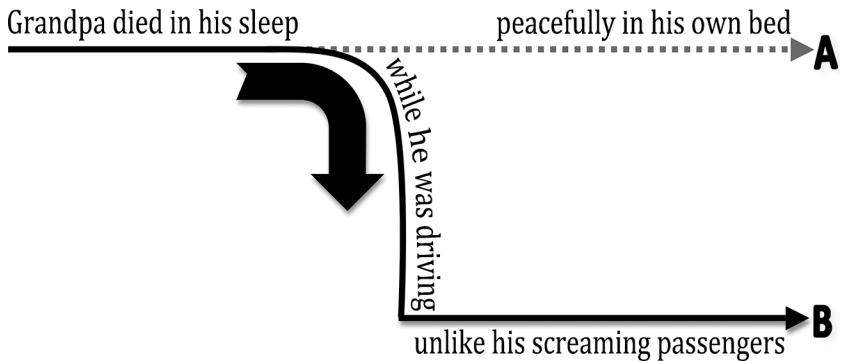


Figure 5.3

The listener's perfectly reasonable assumption of A calls for reinterpretation and an abrupt correction when the comedian instead diverts grandpa along route B.

This setup is itself a cliché, and that is no bad thing. Clichés carry universal truths, of a sort, and can directly tap into the shared experiences of an audience. If we use Toplyn's approach here, we can identify two obvious handles: *gays* and *dressers*. What other cliché unites them both in a single bound? Dean offers the following:

You try spending twenty years in the closet!

Clichés are so firmly entrenched in our networks of human knowledge that beat writers like Burroughs and Gysin embraced extreme methods to avoid them. But comedy shows that we need to embed the same clichés into computer networks of knowledge too. Most clichés are metaphorical in origin, like that of gay people hiding their sexuality in their closets. We need our machines to see these clichés for what they are, but we also want them, when it suits a comedic goal, to take the cliché literally. It takes double vision to go from “good dressers” to “in the closet.”

CREATIVE ACCOUNTING

Max Bialystock has a problem. The great Broadway producer has been humbled by a string of critical bombs, from his modern take on *Hamlet* (*Funny Boy*) to a musical about Einstein's least remarkable sibling (*Young*

Frank Einstein). In the course of having his books done by Leo, a timid accountant, Max takes no comfort in hearing that his recent flop actually made a tiny profit, all because he oversold the show to his backers. With no proceeds to share, nobody can expect to recoup their investment. But Max's heart sings when Leo suggests, as a kind of thought experiment, that it really is possible to make a *lot* of money from a flop, provided it truly is a disaster. Because if Bialystock knows anything, it's how to mount a total clusterflop. With Leo as his partner, he sets out to find the worst play ever.

You'll no doubt recognize this as the plot to the Mel Brooks film *The Producers*, which was later turned into a Broadway show in its own right. Max and Leo face a classic creative dilemma: How do we make something better when everything we try seems to make things worse? For instance, we want our cars to be greener and more fuel efficient, but not if that means making them from flimsy materials that crumple at the first hint of a fender bender. We want more affordable electric cars for the environment's sake, but don't want to spend our lives waiting for them to charge. We want to pay less for air travel, but not if it encourages airlines to treat us like battery chickens. Max and Leo want to make a fortune from a Broadway show, but they have a flair for failure and don't want to invest their own money.

Max and Leo need to uncouple the quality dimension (Will audiences like their show?) from the profit dimension (so they can line their pockets anyway). Their solution goes further still: they manage to invert the relationship between quality and profit so as to ensure a windfall if they put on a truly awful show. Naturally, their ploy goes comically awry, and the show is so awful that it is appreciated as a brilliant satire. Audiences adore it, and so the duo lose their shirts and their freedom. Before they are carted off to jail, in the stage version at least, they sing a duet that is a marvel of humorous script opposition: "Where did we go *right*?"

A good reason to systematize the creative process is to allow us to know in advance where, why, and how we can go right or wrong, so that we can direct our energies toward the former while doing our best to avoid the latter. If we comb through the fossil record of human innovation, as provided by the patent office, we see that many inventions can be characterized by a relationship between two antagonistic qualities. Desirable increases in

one quality, such as speed, capacity, or durability, tend to cause undesirable increases in others, such as cost, weight, size, or waste, or undesirable decreases in positives such as efficiency, usability, or maneuverability. Innovative patents offer us nonobvious ways of balancing the seesaw in our favor, turning a win-lose into a win-win or, at worst, a win-draw.

A wide-ranging analysis of patents can allow us to identify the most frequently opposed qualities and the principles by which inventors have reconciled them. This is exactly what Genrich Altshuller, the inventor of the TRIZ system of creative problem solving, set about doing after years spent toiling in the patent office of the Soviet navy. TRIZ is an acronym for the Russian phrase *Teoriya Resheniya Izobretatelskikh Zadatch*, but its English near-translation—theory of inventive problem solving—also yields a half-decent acronym of its own: TIPS.¹⁵

The antagonistic relationships at the heart of TRIZ/TIPS are the sober-minded equivalents of humorous incongruities. As in a joke, the antagonism often reveals itself as a contradiction that seems to defy meaningful resolution. Take umbrellas: an umbrella is a portable shelter against the rain that is as effective as a tent and just as portable as a cane. Even if our umbrella is wide enough to shelter several people, its folding web of ribbed fabric makes it narrow enough to be carried like a walking stick. The key to resolving the tension between bulk and portability is the recognition that these qualities, although contradictory at the same time, are rarely salient in the same time frame. We need coverage when it rains and ease of carriage when it does not. The apparent incongruity arises from a conflation of these two perspectives and disappears when we carefully pick them apart.

The solution is to give umbrellas a folding structure so they can express scale in one context and compactness in another. Likewise, an umbrella that is firm enough to be used as a shield against the wind can prevent us from seeing where we are going unless we swap the fabric for a transparent plastic that allows us to hold the umbrella in front of us *and* see through it at the same time. An umbrella that is so compact that it fits in our pocket can unfold laterally *and* vertically so as to give us the same coverage. In yet another innovation that its inventors didn't care to patent, the Bulgarian secret service turned an innocuous umbrella into a weapon befitting James

Bond. On a London bridge in 1978, they assassinated the defector Georgi Ivanov Markov with an umbrella that had been modified to shoot a tiny ricin pellet into its target's leg. Luckily, this odd job was motivated by a very rare incongruity: banality versus lethality.

In his search of the patent records, Altshuller identified thirty-nine kinds of incongruity and forty different principles for resolving them. In the bittersweet world of TRIZ, each incongruity pairs a quality that an obvious solution will improve to one that it will degrade. Making cars heavier to make them more robust also makes them slower and less fuel efficient, but the converse is also true: lightweight materials improve speed and fuel efficiency while reducing robustness. Altshuller compiled a 39×39 matrix for all possible pairings of his thirty-nine qualities, using rows to designate the improving qualities and columns to designate the worsening ones. Since every TRIZ incongruity maps to a unique cell in this matrix, Altshuller placed numeric references to one or more of his forty TRIZ design principles into each cell. Look up an incongruity, and the matrix will offer you a way out of your dilemma.¹⁶

Suppose we want to increase the speed of an electric car, but not at the cost of decreased battery life. The improving quality is *speed*, while the one that worsens is *use of energy by moving*. If we look up the TRIZ matrix at the intersection of this row and this column, we find a reference to several inventive principles. Principle 8, Antiweight, suggests we reduce the weight of the moving object by introducing lift, since an aerodynamic shape can reduce drag and energy expenditure at high speeds. Principle 18, Dynamics, encourages us to make our design more adaptive to its context, so as our electric car moves faster, its shape might alter, perhaps by unfolding a rear spoiler. Using principle 35, Parameter Changes, we might achieve the same end with something as mundane as an automatic transmission.

Have you ever looked into the engine of a jet plane as you embark via the front steps? Most jet engines have a perfectly circular cowling around an inner turbine. The thrust of an engine is related to how much air it can suck in and compress, so increasing the radius of the cowlings will increase the air intake to the turbines. However, this increase can

dangerously reduce an engine's ground clearance. So how *do* we increase the surface area of the cowling without increasing its radius? The consultants who advised Boeing on their 737–400 engine suggested a change that now looks rather obvious: jettison the belief that cowlings must be circular.

You may have noticed that some Boeing engines have a bun shape that is flat at the bottom to preserve ground clearance and wider at the sides to increase air intake.¹⁷ The consultants arrived at this counter-circular insight using principle 4, Asymmetry. Table 5.2 provides a quick inventory of all forty TRIZ principles:

Table 5.2.

The 40 design principles of TRIZ.

1. Segmentation	21. Speed Skipping
2. Taking Out	22. Blessing in Disguise
3. Local Quality	23. Feedback
4. Asymmetry	24. Intermediary
5. Merging	25. Self-Service
6. Universality	26. Copying
7. Nested Doll	27. Cheap Short-Lived Objects
8. Anti-Weight	28. Mechanics Substitution
9. Preliminary Anti-Action	29. Pneumatics and Hydraulics
10. Preliminary Action	30. Flexible Shells and Thin Films
11. Beforehand Cushioning	31. Porous Materials
12. Equipotentiality	32. Color Changes
13. The Other Way Round	33. Homogeneity
14. Spheroidality/Curvature	34. Discarding and Recovering
15. Dynamics	35. Parameter Changes
16. Partial/Excessive Actions	36. Parameter Changes
17. Add Another Dimension	37. Thermal Expansion
18. Mechanical Vibration	38. Strong Oxidants
19. Periodic Action	39. Inert Atmosphere
20. Continuity of Useful Action	40. Composite Materials

If we treat each quality as a handle, then Altshuller's matrix gives us an inventory of topic associations, where the numbers in each cell suggest the TRIZ equivalent of Toplyn's angles. We just need to identify the handles for our specific situation, look up the matrix, and read off the angles that have led to past success. To see how we might deliberately produce humorous results in this way, consider a joke written for the Jimmy Fallon show that Toplyn discusses in his book:

A company in Seattle just came out with a new bacon-flavored soda. So if you love the taste of bacon and you love the taste of soda, you're about to realize how much you love them separately.

The joke hinges on a clear script opposition: we expect it to end with *together*, but the writers opt for *separately* to reveal their distaste for the blend. By laughing at the blend's ingredients, the joke also subverts principle 40, Composite Materials. Two related TRIZ principles—Merging (principle 5) and Universality (principle 6)—also provide the spark for the following joke, written for the *Late Late Show* with James Corden:

Amazon is working on a robot that follows its owner around the house. It's basically an Alexa on wheels. . . . Here's how you get one: you leave your Alexa and your Roomba alone in a bedroom and let nature take its course.

This joke blend applies the Merging principle to imagine how one object can be turned into another, and uses the Universality principle to show how the resulting blend can fulfill the functions of multiple objects at once. Although the blend is not funny in itself, it does provide the rationale for the oppositions at the heart of the joke: sex versus innocence, artificial versus real, and modern versus old-fashioned.

In the world of design, TRIZ principles are used to fully resolve contradictions so that the resulting product is surprising but not risible. In contrast, when TRIZ principles are used for jokes, the incongruity is rarely fully resolved. The purpose, after all, is to make audiences laugh at a novel idea, not for them to embrace it as a serious innovation. Fallon and Corden make fun of other people's ideas, but when the idea is the comic's own, a careful framing is needed to achieve ironic distance.

Consider this particular joke from Adam Bloom at the 2004 Edinburgh Festival Fringe:

Is it fair to say that there'd be less litter if blind people were given pointed sticks?

Bloom questions his own motives so as to prevent his joke from appearing too insensitive. By contrast, the following joke, written for the Conan O'Brien show, projects this insensitivity onto the imaginary actions of big business:

In a fashion show that took place yesterday, Dolce & Gabbana sent their handbags down the runway on drones instead of models. But first, each drone was forced to lose 10 pounds.

This joke can be understood in terms of several TRIZ principles. For instance, the phrase “But first” evokes principle 9, Preliminary Anti-Action, in which a predicted problem is mitigated in advance, such as by swallowing antacids before eating a pepperoni pizza. It is a cliché that models are routinely asked to lose weight, and this concern with weight is salient in the drone frame too. This suggests principle 8, Anti-Weight, although we can also view the replacement of human models as an inversion of principle 28, Mechanics Substitution. Pairing principles 8 and 28, we might also imagine that the drones lose weight not by mechanical means, but by using proven fashion industry solutions: dieting, liposuction, and appetite suppressants.

A number of TRIZ principles also have clear parallels with logical mechanisms that have been studied in the humor literature. Take principle 13, The Other Way Round, which corresponds to a number of logical mechanisms that fall under the rubric of *Reversal*; these include role reversal (in which event participants swap roles) and figure-ground reversal (where a foreground element is backgrounded, and vice versa¹⁸) as well as the resigned humor of vacuous reversal (in which a script’s event sequence is inverted, but the change has little or no effect). Consider the use of role reversal in this joke, written for *Late Night with Seth Meyers*:

A team of scientists recently completed an experiment studying the effects of the drug MDMA on octopuses. Which is part of a bigger experiment of what happens when you give scientists LSD.

The value of TRIZ as means of humor production lies not in its contradictions and principles, but in the matrix that maps from one to the other. If we can find a pair of salient qualities in a joke setup, we can look up the matrix entry for the pair to find specific guidance on which principles to consider. The qualities on which TRIZ focuses are chosen to be representative of the innovations that cross the desks of patent officers, but these are not so very removed from the concerns of a typical joke. It helps that TRIZ's focal qualities emphasize aspects that easily lend themselves to innuendo and "double vision" when they are applied to people, such as weight, size, volume, shape, speed, force, temperature, pressure, stability, stress, energy, intensity, versatility, durability, and reliability.

So let's apply some double vision to the properties that relate to Conan's drone joke. Since each drone carries a handbag, we might focus on the property *weight of moving object*. Since each replaces a fashion model, we might consider how this weight will affect its *speed*. The TRIZ matrix offers four principles for this pairing: Taking Out (2), Anti-Weight (8), Dynamics (18), and Strong Oxidants (38). In the spirit of principle 2, we might imagine that each drone throws up backstage, before take-off. For principle 8, we might imagine that each drone ditches its baggage, or crash-lands, Sully-style, into a chocolate fountain. For principle 18, we can explore ways in which each drone adapts to its role as a fashion model, perhaps by draping it in silk or covering it in bling. Finally, for principle 38, we might imagine supercharging the drone's fuel with Red Bull and diet pills. While we are unlikely to use any of these possibilities in its raw form, one might still be reworked to produce a joke as polished as Conan's own.

SOME ACCESSORIES SOLD SEPARATELY

Schematic approaches can take us only so far, whether they happen to be based on theory, data analysis, or hard-won experience. Their generic schemas can guide our thoughts and structure our thinking, but they cannot do our thinking for us. Even if, like fancy food processors, they offer a wealth of attachments and settings, it still falls to us to bring the ingredients: our knowledge of words and the world. Fortunately, the topical setup to a joke,

such as a headline or a factoid, can provide some of the key ingredients, and guide our choice of the rest. Consider this setup:

A new poll has found that a majority of Americans say that driverless **cars** will have a big impact on the **elderly**.

In addition to the most obvious handles, which are highlighted in bold, the setup also points to an effective angle for a joke. The word *impact* has two meanings with opposing positive/negative connotations: a significant effect and a physical collision. This is how writers for Seth Meyers turned this pair into a punch line:

Specifically, if they don't cross the street fast enough.

It is our knowledge of words (the double meanings of *impact* and their differing connotations) and of the world (the elderly tend to be slower and more fragile), not the schematic elements of our technique, that allows us to unite both frames.

Schematic approaches can be applied in forward mode, to help us invent, or in backward mode, to analyze someone else's inventions.¹⁹ Indeed, the input to one mode is often the output of another. Look at the jokes that find favor with late-night hosts, and you will see quite a few that bear the hallmarks of a system like TRIZ. This strange offering bears the mark of TRIZ principle 32, Color Changes:

The crayon company Crayola has launched a new line of makeup based on its crayon colors.

Crayola isn't actually selling crayons that can be used as makeup, which would be the result of using principle 5, Merging, or principle 6, Universality, but it is not a stretch to imagine that they are. Kids have been raiding mommy's makeup box for as long as lipsticks have looked like crayons, so Crayola's new idea might also be seen as an application of TRIZ principle 13, Other Way Around. In these contexts, principle 22, Blessing in Disguise, asks us to think of crayon cons that are makeup pros, or vice versa. Kids don't mind when they color outside the lines, but adults do. So this is how James Corden's writers turned this difference into a punch line:

People who've tried the Crayola makeup say the colors are great, but they did have trouble staying inside the lines.

This punch line might also be created by following Joe Toplyn's advice for writing jokes based on a new product: "Ask yourself, 'what are people saying about it?'" Little kids, who sometimes eat their crayons, are people too, so what are they saying? Refining the original punch line to apply it directly to kids can give us this variant:

Kids who've tried the Crayola makeup say the colors are great, but they just don't taste the same.

Coincidentally, even the product announcement seems to resonate with this idea:

Crayola says their new line of makeup is completely vegan.

Presumably this means that Crayola makeup is not tested on animals, but comedy is rarely hailed for its generosity of interpretation, so Corden's writers give us:

Wait, so are you telling me the makeup I've been eating has meat in it?

Two opposing senses of "vegan"—that is, "food from nonanimal sources" versus "products that cause no harm to animals"—compete for our attention in the joke setup. These opposing senses, with their own frames and scripts, then collide in the punch line in a flash of educated ignorance. Nonetheless, while the mechanical switch from one sense to another is necessary for the humor, it is not its actual cause. That has more to do with the emotional change that the switch triggers in the audience. The food sense allows a comic to undercut the grand philosophical sense with the silly image of a grown man eating makeup. By making fun of a topic, a joke can alter our emotional response to it. If we think that vegan products and those who buy them are a tad pretentious, Corden's joke allows us to feel a tad superior too. In the end, then, no matter what mechanisms we use to ram two frames or worldviews together, they can result in humor only when the collision engages our emotions.

FEED THE HOLE

Principles are for serious people. There is nothing remotely humorous about any of the forty principles in TRIZ. Neither is anything humorous about the thirty-nine qualities in which the framework's contradictions are rooted. Yet, for that matter, there isn't anything inherently funny about any of the oppositions in humor theories such as the semantic script theory or the ontological script theory of humor, or, indeed, about any of the logical mechanisms that have been identified.

The mystery of humor as a process, whether psychological or computational, is that it arises from a marriage of unfunny elements. TRIZ's principles and qualities and contradictions, much like the oppositions and logical mechanisms of humor theories, are simply a means of exploring a possibility space. Burroughs and Gysin used a scissors to explore the space of possible texts in their search for something new and surprising, and AI can use these formal structures to do the same. The value of a framework like TRIZ, even though it is not intended for humor, lies in its systematicity. It doesn't solve any mysteries, but it is still a useful tool, one that allows a comedian to explore a space in search of emotion-altering possibilities.

When Seinfeld pictured Carlin as "a train hobo with a chicken bone," he didn't imagine the natural comic using a systematic technique like TRIZ. But feeding the great maw of television's topical entertainment shows requires a steady stream of disposable jokes that must be cranked out on an industrial scale. Joke assembly lines can no more afford blockages than a fast food assembly line can, so a systematic approach to production is crucial for both. If these systematic processes allow comedy writers to fake it 'til they make it—and for AI, making and faking may amount to the same thing—then so be it. The matrix will see you now.

6 DANGER, DANGER: INCONGRUITY AND THE TIME COURSE OF JOKES

THE NAKED TRUTH

Contradictions are fatal to any scientific theory that aims for generality, but we humans handle inconsistencies in our stride. In a confessional essay for *Esquire*, aptly titled “The Crack Up,” F. Scott Fitzgerald famously wrote that “the test of a first-rate intelligence is the ability to hold two opposed ideas in mind at the same time and still retain the ability to function.”¹ Fitzgerald’s observation presupposes a certain detachment from these opposing ideas, as though he were weighing the merits of two melons in a supermarket, but it takes detachment like this to step outside two competing frames of reference and find the insight that can unify them both.

We can call these frames *matrices*, as in Arthur Koestler’s theory of bisociation, or *mental spaces*, as in Mark Turner’s and Gilles Fauconnier’s theory of conceptual blending, or *scripts*, as in Victor Raskin’s semantic script theory of humor. The nomenclature matters much less than the realization that we humans excel at compartmentalization and can easily hold two opposing ideas in different parts of our minds so as to hold them both true at the same time. So we separate sexual appetite from religious belief when the latter abhors the former, or preach with more conviction than we practice, or tolerate aspects of ourselves that we condemn in others. We embrace contradictions without resolving them because we reason locally as well as globally, and we erect mental barriers between opposing ideas. This compartmentalization prevents an incongruity in one set of beliefs from destabilizing our entire worldview, like cracks radiating from a chip

in a windshield. Jokes alert us to the existence of the barriers between compartments, while localizing the humorous conflict to beliefs on opposite sides of a boundary.²

So what would it be like to have no barriers at all, so that our thoughts could encompass everything we know and believe all at once? We can get a good sense of this from science fiction, where human-like AIs crack up all the time. We relate to fictional supercomputers on an emotional level, just as they seem to do to us. We feel awe for them, yet they—despite disavowing all emotion—act superior to us. Their claims of logical superiority sound awfully like boasting, so when their logic leads them astray, it's only natural that we experience our own jolt of Hobbesian superiority. In TV's original *Star Trek*, this logic-versus-emotion theme played out every week in the playful back-and-forths between Captain Kirk and Mr. Spock and between Spock and the ship's doctor, "Bones" McCoy. But it would reach its zenith whenever Kirk took a mental monkey wrench to an overbearing computer.

In an episode titled "The Ultimate Computer," Kirk outwits M-5, a murderous AI that was naively designed to offer military guidance to the *Enterprise*.³ When the M-5 shows that it sees no difference between strategic war games and real warfare, Kirk reminds it that the killing of redshirts is inconsistent with its prime directive to protect lives. Confronted with the incongruity of its actions, the M-5 shuts itself down in an act of logical expiation. In "The Changeling," Kirk crosses circuits with Nomad, an intelligent space probe that mistakes Kirk for its creator, Jackson Roykirk. Nomad's mission is to sterilize imperfections wherever it finds them, so Kirk turns this mistake against it.⁴ As he is in fact *Kirk* and not *Roykirk*, the machine has erred; as it has not recognized its error, it has erred again; and as it has not sterilized itself for its errors, it has erred once more. Kirk's guilt-trip is enough to send Nomad into a spiral of self-recrimination and eventual implosion.

Machines that extol the superiority of their own reasoning may be asking for a comeuppance, but this doesn't always cause feelings of "sudden glory" (as Hobbes put it) in the humans that outsmart them. The HAL 9000 computer of Stanley Kubrick's *2001: A Space Odyssey* has no time for false modesty when he says, "It just happens to be an unalterable fact that

I am incapable of being wrong,” yet there isn’t a dry eye in the house when HAL is finally undone by the only human he has failed to murder.⁵ Perhaps it’s because we really feel for HAL, and see his actions as the effects of a crack-up as profound as that reported by Fitzgerald, that we sympathize with his sudden realization that he is not the infallible machine he once believed himself to be. Poor HAL has been given conflicting orders by his human programmers: to be open and friendly in his relations with his crew and to defend the integrity of his mission at any cost. The latter requires him to lie to those crewmates who are not privy to the mission’s mysterious objectives, and although his crack-up starts small, it’s not long before HAL is cleaning house.

The first tiny cracks in his logical facade appear when HAL plays chess with a human crewmate, Frank Poole, but are so subtle as to resemble a screenwriting gaffe.⁶ HAL, playing black, sacrifices his queen to lure Frank’s bishop into a trap. Once the bishop is taken by HAL’s knight, Frank’s king will be in checkmate. HAL describes his setup move as “queen to bishop three,” but this board reference is incorrect. From HAL’s side of the board, the move should be described as “queen to bishop six,” since the queen is pressing deep into Frank’s territory. Remarkably, HAL seems to be describing his move from Frank’s viewpoint, not his own. Is this just a case of HAL (or the meticulous Kubrick) fluffing his lines, or does it reveal that HAL secretly identifies with the humans from whom he must hide key data? In any case, it is not long before HAL is luring Frank outside the ship with a setup that resembles his queen sacrifice—he pretends an antenna is on the blink—and he soon graduates from sacrificing chess pieces to sacrificing crewmates.

HAL is the possessor of a holographic memory, so he maintains a holistic, noncompartmentalized view of all of his knowledge and beliefs. The little home care robot in the 2012 movie *Robot & Frank* describes this model of memory storage rather well: “My memory is a holographic array. If I lost half of it I’d still have every memory, just in half the resolution.”⁷ This prevents the robot, or HAL, from doing what Frank naively asks of it: “Can’t you just erase the bad parts?” When HAL’s storage units are removed by the crew’s sole survivor, we see—or rather *hear*—him slowly dissolve into nothingness as he wistfully sings the song “Daisy Bell.” With lumps in

our throats, we appreciate why a single incongruity in HAL's programming could have such pervasive, and devastating, consequences.

It doesn't pay to put all our eggs, or memories, in one basket, and it certainly doesn't make compartmentalization any easier if we scramble them first. This is the lesson of Zero, the supercomputer that serves as "the world's filing cabinet" in the 1975 film *Rollerball*.⁸ Zero's memory also works along holographic lines, an approach that is not without its drawbacks. As a curator puts it, "He's the central brain, the world's brain. Fluid mechanics, *fluidics*. He's liquid, you see. His waters touch all knowledge. Everything we ask has become so complicated now." Zero has made printed books and physical libraries obsolete, so humans must look to the machine for answers to just about every question of consequence. When a Rollerball player of global renown, Jonathan E, comes to question Zero about the nature of corporate decision making in this future dystopia, Zero's response is:

Answer. Corporate decisions are made by corporate executives. Knowledge converts to power. Energy equals genius. Genius is energy. Corporate entities control elements of economic life, technology, capital, labor and markets. Corporate decisions are made by corporate executives. Negative, negative, negative . . .

Zero is plainly mad. Its memory is so overburdened by the weight of the world's knowledge that it has collapsed into itself, much the same way a collapsing star forms a singularity. All that remains visible on its event horizon are tautologies and dense metaphors. While jokes and metaphors show an "educated insolence" for the boundaries that separate belief spaces, they also rely on those boundaries to work their magic. So it doesn't help if those boundaries disappear under the weight of the beliefs they are supposed to corral. But this at least takes us some way toward an explanation of Zero's name: a machine that knows everything—every detail, every nuance, and every fractured and irreconcilable perspective—must be so insanely conflicted and ambiguous that it cannot assert anything with certainty. Drowning in a sea of cavils and caveats, Zero has good reason to screech "Negative" to every query. As its poor, exasperated curator puts it, "He considers everything. He's become so ambiguous now, as if he knows nothing at all."

The prognosis for fictional AIs with a superiority complex is rarely promising. Either they crack up under the strain of preexisting incongruities or they are brought low by the incongruous pseudo-logic of a destabilizing agent. The latter has never been so well expressed as in Jean-Luc Godard's 1960 art house film, *Alphaville*.⁹ The city of Alphaville is ruled with an iron hand by a supercomputer named Alpha 60 that burnishes its logical credentials by lecturing others like a French semiotician. Lemy Cau-tion, the grizzled agent who is sent to end the AI's reign, is not impressed and does not mince words when he tells Alpha 60 to "fuck yourself with your logic." This is exactly what the machine does: it screws *itself*. More accurately, it is the antilogic of a riddle about true love that turns the logical Alpha 60 into a gibbering wreck (this is an art house film, after all). In a sense, all humorous incongruities are invitations to "fuck yourself with your logic," albeit typically with more entertaining and less dramatic results. By urging us to apply conventional wisdom to unfamiliar edge cases that transcend the conventional, jokes can show us, and even make us feel, the limits of commonsense reasoning.

Fortunately, we humans are not so susceptible to all-consuming incongruities. Whenever *we* are conflicted, we can simply choose to believe or not to believe, insisting that others "don't go there!" when our barriers are in risk of crumbling (or, seeing other people's barriers for what they are, we might go Full Metal Jack Nicholson and shout, "You can't handle the TRUTH!"). Logical contradictions don't make us "crack like an old plate," to borrow a phrase from Fitzgerald's essay, but they can still be fatal to the human institutions that are supposed to operate with internal consistency—in other words, like a well-oiled machine—such as the judicial system. In the logical crack-up of these institutions, we can see something of how machines are expected to respond to incongruity. In his book on rhetoric, *You Talkin' to Me?*, Sam Leith recounts an apocryphal case of mutually assured incongruity from the courts of ancient Syracuse that is worth retelling here.¹⁰

Leith tells of a contractual dispute between two rhetoricians, Corax and Tisias. Corax was a teacher of rhetoric to aspiring young lawyers, one of whom, Tisias, could not afford his fees. They agreed to an arrangement whereby Tisias would pay Corax only if his new skills allowed him to win his first case. But Tisias was a crafty one and never took a case all the way

to trial, preferring instead to settle out of court. With no first case to win or lose, Tisias thrived while Corax went without his fee. So, the master dragged the student into court, believing himself to have an airtight case. In Corax's view, the judge was bound to decide in his favor since only two outcomes were logically possible: if Corax won, he should receive his requested costs; but even if he lost, Tisias would have won *his* first case and would still owe Corax his fee. Yet Tisias also felt his case to be a forgone logical conclusion, for if Tisias won, he would owe Corax nothing, and if Corax won, the student would have lost his first case and so would still owe his master nothing.

The judge, perched on the horns of a mathematical dilemma, needed to weigh the demands for consistency and completeness. To be consistent, the law cannot tolerate contradiction, so his judgment could not conflict with itself or with legal precedent. To be complete, the law must be capable of rendering judgment in every case, even if doing so gives rise to a legal contradiction. In the case of Corax and Tisias, contradiction is inevitable: each holds a winning hand, but they cannot both win or both would also lose. Yet if we are forced to choose, incompleteness is preferable to inconsistency. An inconsistent judgment is a bad precedent from which a deluge of future contradictions can flow, allowing litigants to build sound arguments from nonsense. An incomplete legal system admits that there are some cases it is not capable of judging, for fear of rendering itself worthless as a whole. In *Corax v. Tisias*, the judge set a precedent for future mathematicians to follow. Choosing consistency over completeness, he kicked them both out of court.

Mathematicians have long sought to place mathematics on a sound, systematic footing, and in the early twentieth century, they set themselves a bold goal: to find a generative proof procedure capable of generating all the theorems that can be formally derived from a given set of axioms.¹¹ This procedure could then be used to crank out proofs for or against any conjecture that had previously eluded their best efforts. However, a young Austrian mathematician, Kurt Gödel, derailed the whole enterprise with a momentous proof.¹² He showed that every formal system of a certain expressive power must also contain a hidden fault line, called the *Gödel statement*, whose truth cannot be established within the system itself. Like Tisias's contract, a Gödel statement refers to itself and to the system in

which it is constructed. It's the kind of statement that a system should be able to prove, yet Gödel showed that it cannot do so without surrendering its consistency or its completeness. The statement is a joke at the expense of the system that houses it.

Gödel proved that formal systems can tie themselves in knots when they are expressive enough to make their own workings the subject of their analyses.¹³ His proof shows that such systems necessarily entail truths that they themselves can never prove. We are forced to accept incompleteness to avoid logical incongruity. Gödel's proof is often cited as evidence that formal systems, whether as abstract algorithms or real machines, are fundamentally unable to reach the bar set by the human intellect. His proof can be taken as a mathematical expression of Bergson's view that machines are too rigid to match the agility of the human spirit unless we turn ourselves into machines too. Scholars of Gödel's work, such as Ernest Nagel and James Newman, have used Gödelian arguments to refute the possibility of real artificial intelligence.¹⁴ However, in his edited edition of Nagel and Newman's classic work, *Gödel's Proof*, Douglas Hofstadter offers the following rebuttal:

Although computers, as their name implies, are built of rigidly arithmetic-respecting hardware, nothing in their design links them inseparably to mathematical truth. It is no harder to get a computer to print out scads of false calculations ("2 + 2 = 5; 0/0 = 43", etc.) than to print out theorems in a formal system. A subtler challenge would be to devise 'a fixed set of directives' by which a computer might explore the world of mathematical ideas (not just strings of mathematical symbols), guided by visual imagery, the associative patterns linking concepts, and the intuitive processes of guesswork, analogy, and aesthetic choice that every mathematician uses.¹⁵

In his defense of machines, Hofstadter ascribes to human mathematicians many of the same qualities we ascribe to comedians: patterns of conceptual association (as in scripts and frames), intuition, guesswork, analogy, and choice, all guided by aesthetic considerations of what works, what doesn't, what amuses, and what repels.¹⁶ More significantly, he leaves the door open for machines to come in and lay claim to these qualities for themselves. Shrewdly peeling away sincerity from truth, Hofstadter

recognizes that a machine that is bound by its formal model of the truth isn't also bound to speak only the truth as it sees it. If this seems a tad cynical, here is *Interstellar*'s TARS to explain why it might also make good sense:

Cooper: Hey TARS, what's your honesty parameter?

TARS: 90 percent.

Cooper: 90 percent?

TARS: Absolute honesty isn't always the most diplomatic or the safest form of communication with emotional beings.

Cooper: Okay, 90 percent it is.

This is what it means to bend the truth: not to alter the truth itself but to pull on the elastic bands that tie words to meanings and facts to each other. This elasticity buffers us from the most dramatic effects of logical inconsistency and incongruity. It allows us to equivocate our way out of potential paradoxes and even to create them for others. Jokes and incongruities of all kinds may strain our expectations, but they do not break us in the process. We are not the hopelessly rigid machines of classic sci-fi movies and TV shows. Thankfully, neither are our computers.

JOLTS AND VOLTS

Kant viewed incongruity as a strained expectation, so it's perhaps apt to think of the herniating about-face of a joke that first turns us one way and then the other as the mental equivalent of a groin strain. The mathematician John Allen Paulos suggests a more formal way of thinking about this bait and switch, and asks us to instead think of jokes as surfaces in an abstract space on which we can explore paths that are more *and* less traveled.¹⁷ These surfaces conceal gaping chasms—the incongruities at the heart of jokes—that we are unlikely to see until it's too late. While the path less traveled is the safer choice and takes us around and not into the chasm, it is also the least attractive. In script terms, the longer path is the script we should be following if we weren't so busy walking off a cliff.

Paulos argues that an area of mathematics called catastrophe theory offers the best tools for understanding and modeling the warped surfaces that contain these beguiling paths and sudden discontinuities.¹⁸ We can model the geometry of jokes using some basic origami, and we don't even need scissors. First, take a flat sheet of paper between your hands and gently bend it into an arch. This bend creates a curved surface, or manifold, that hides no surprises for any explorer that might amble across it. Now pinch one side of this arch into a kink, being careful not to crease the paper. This new manifold is a three-dimensional "cusp." Let's follow an imaginary ant as it walks across the surface. As it crosses the cusp, the ant will temporarily reverse its direction of travel and move forward again only when it is clear of the cusp. When seen side-on in a two-dimensional plot, the ant's path will resemble a hairpin bend, but when viewed from above, its path will look like the image in figure 6.1.

As a natural wall crawler, our imaginary ant needn't worry about falling over the edge of the cusp and might not even notice the fold that impedes its progress. We, however, would suffer a dramatic fall if we naively ambled over this treacherous dip. Let's look at two competing paths along the surface, A and B (figure 6.2). Path B leads the wanderer around the cusp, while path A leads the wanderer *over* the cusp:

It is reassuring to see that when it comes to jokes, the intuitions of a comedian and a mathematician are perfectly compatible. Greg Dean's view

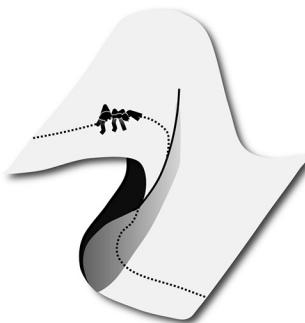


Figure 6.1

An ant traverses a sheet of paper that contains a three-dimensional cusp.

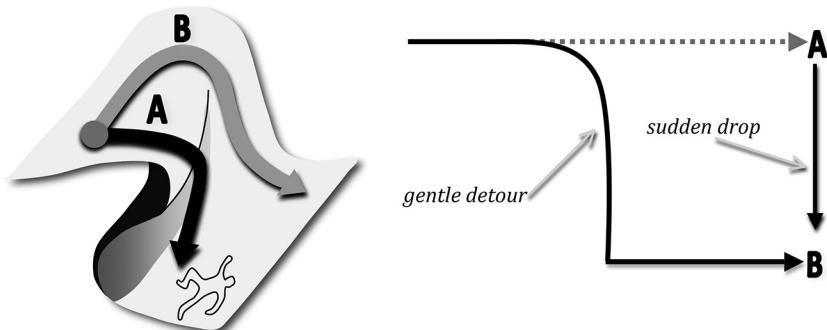


Figure 6.2

Two paths along our abstract surface. The longer path, B, avoids the sudden drop. Notice how the surface projects the analysis of the last chapter into three dimensions.

of joke structure from the previous chapter can be understood as a 2D projection of the cusped 3D surface proposed by John Allen Paulos. From our initial vantage point, this cusp looks like a gentle dip, not a precipitous drop. Alas, since the gentler gradient of B is purchased with a sideways detour, it is A that promises the most direct route to the edge of the manifold. When A and B are viewed as geometric representations of competing interpretation strategies, we can appreciate why a gung-ho explorer would prefer A and switch to B only when faced with a sudden incongruity. In fact, it's tempting to think of A and B as distinct personality types. People of type A are guided by gut feeling and intuition and rush headlong to obvious conclusions. People of type B are detail-oriented and methodical. Slower but surer, they reach their conclusions after people of type A have already reached theirs, but they can explain their reasoning and show why their answers must be the right ones.

As personality types, A and B are caricatures painted with the boldest strokes, so we shouldn't expect any individual to conform to just one. There are shades of A and B in all of us, and it is more useful to think of them not as distinct types, but as thought processes that we all possess. According to dual process theory, an influential account of cognitive function, A and B are two competing processes for dealing with whatever the world throws at

us.¹⁹ Process A is rapid, responsive, and always on, guided as it is by personal experience, familiar stereotypes, and deep-seated associations. Process B employs a purer form of reasoning that works from first principles. Since B is more demanding of our mental energies, it is slower to engage with a problem, slower to suggest answers, and more easily depleted.

In his book *Thinking Fast and Slow*, the psychologist Daniel Kahneman labels the A and B facets as *system 1* (fast but reckless) and *system 2* (slow but careful).²⁰ System 1 is a hare, swift and responsive, always active and ready to race, as when it eagerly hurtles down path A. The unconscious priming of associated ideas is a key feature of system 1. When these automatic processes work to our advantage, it is system 1 that greases the rails of intuition. But when they work against us, causing us to fixate on red herrings and near-misses, system 1 makes the distractions hard to ignore. In chapter 1 we briefly touched on a joke from a fictional AI called Mike. In his role as colony administrator, Mike has issued an astronomical paycheck to a janitor, safe in the knowledge that it is just too large to ever be cashed. Now place yourself in the headspace of this employee, just as Mike surely did when he formulated his prank. Your inner system 1 shouts, “I’m rich!” and has you reaching for the phone to make a screw-you call to the boss, before system 2 swoops in with the awful truth: the check is a dud, and you are still poor.

System 2 may be surer and its insights may be more reliable, but in the race to conclusions, system 1 almost always beats it to the punch. We need to engage the good offices of system 2 whenever system 1 jumps too quickly to a conclusion that leaves us feeling stymied. Although system 2 often cleans up after system 1, it has little direct control over system 1: it can step in to repair the latter’s mistakes, but it cannot stop those mistakes from occurring in the first place. Yet for all that, the occasional failings of system 1 should be a cause for celebration, not concern. Like Mike, comedians use their understanding of system 1’s shortcomings—no doubt derived from personal experience—to entertainingly exploit these failings in the system 1 processes of other people.²¹ A well-crafted punch line exploits the rift between systems 1 and 2. Without this divergence,

which is essentially a tale of two velocities and two arrival times, there can be very few jokes of the two-stage blunder-and-recover variety that are so popular in the humor literature.²²

When systems 1 and 2 work in harmony to reach the same result, we have the satisfaction of knowing that our intuitions are backed by reason. But when they disagree, the confusion of different answers arriving at different times results in what computer scientists call a “race condition.”²³ This is the logical equivalent of a comedic double-take, leading a system to act on the basis of one truth value and then another, causing a long line of other dominoes to dramatically fall over too. A logic probe can help us to see the effects of race conditions and track them to their source. When systems 1 and 2 cause race conditions of their own, as when jokes use the familiar to lure us over the edge of common sense, we can hope that the brain also signposts our ensuing stumbles with a measurable change. Fortunately, just as a logic probe can register a sudden spike in current or voltage in a specific part of a circuit, our brains register the confusion of race conditions with spikes of their own, ones we can measure in surprisingly simple and noninvasive ways.

The invention of the phonograph revealed a remarkable insight about music: no matter how culturally diverse or tonally complex the source, any music at all can be captured by the movements of a needle scratching its way along a smooth recording surface.²⁴ The human brain that is responsible for all this diversity and sophistication is itself staggeringly complex, yet we can also capture some of its dominant notes with the movement of a single needle. The recording won’t have the lushness of an LP, but we can still hear enough of the music to figure out what kind of instrument is being played. In this way, an electroencephalogram (EEG) gives neuroscientists a noninvasive way to eavesdrop on the cycles of the brain.²⁵ By attaching electrodes to the scalp and aggregating the voltage at different areas, we can plot the brain’s changing electrical activity on a scrolling sheet of paper. Of course, we will need to average the readings over a range of subjects for the same stimuli if EEGs are going to draw us a general picture of the brain’s responses to a specific kind of load, as when subjects are faced with a humorous incongruity.

If the readout scrolls from left to right, then the horizontal dimension is time, as measured in milliseconds. The vertical is electrical potential, as

measured in microvolts. Since this axis is commonly inverted, a transitory deflection toward a negative potential draws a peak on the page, while a lurch to the positive draws a trough. These aggregate peaks and troughs are components of an ERP (event-related potential), a chart of the brain's electrical activity in the second or so after a stimulus is presented to subjects.²⁶ The negative deflection in voltage that peaks 100 ms or so after the stimulus onset is named N100, and the positive deflection that bottoms out at around 200 ms is named P200. Since the brain must figure out what kind of stimulus it is dealing with before it can grasp its meaning, earlier ERP components tend to relate more to sensory processing and later components relate more to semantic analysis.²⁷ When the stimulus is a word, a text, or an image that requires us to reason about its meaning, the N400—a negative deflection peaking about 400 ms after onset of the stimulus, as shown in figure 6.3—is the ERP component with the most to tell us about humor.²⁸

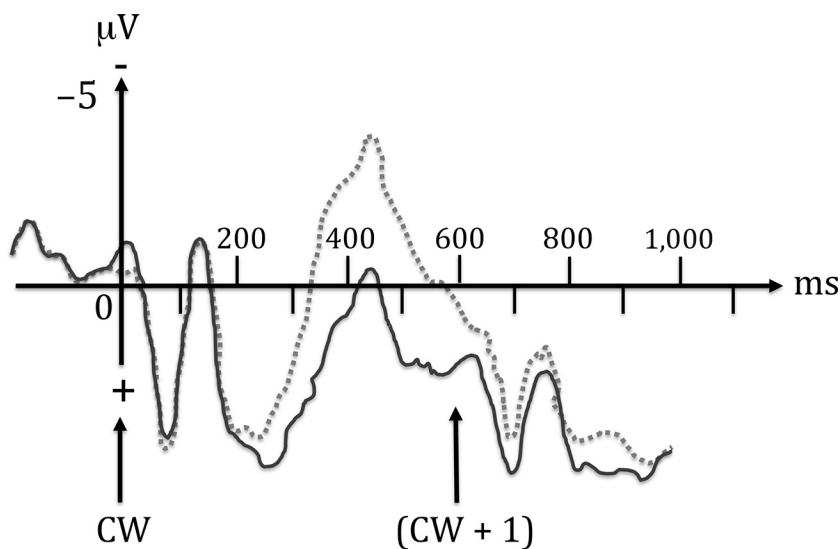


Figure 6.3

Average N400 negative peak after onset of the critical word (CW) in a sentence. Continuous line = coherent sentence; dashed line = sentence with unexpected word.

Source: This figure is based on van Berkum, Hagoort, and Brown, "Semantic Integration," 660.

We don't expect the N400 to form much of a peak when a stimulus ends with a word that accords well with what has gone before (continuous line in figure 6.3). But when it bows in a surprising finale that forces us to rethink our expectations, as in a witty one-liner (dashed line in figure 6.3), we may detect a clear N400 spike. Consider these stimuli:

1. I went to my favorite restaurant and ordered a pizza to *go*
2. I went to my favorite restaurant and ordered a pizza to *leave*
3. I went to my favorite restaurant and ordered a pizza to *dance*

The first stimulus ends with an especially congruent word choice: we order food "to go" from a restaurant, and pizza is a popular take-out option. The second stimulus looks almost as if we've chosen "leave" as a synonym for "go," but the choice is incongruous in this context, suggesting that we went into the restaurant just to bully the menu and evict the Italian option. While we can be charitable and assume that the speaker means, "I ordered a pizza to leave [with]," this mind-set can't do much for stimulus 3, which poses an incongruity that resists resolution.

Each stimulus is constructed so that the critical word occurs at the end. So when we compare ERPs for all three stimuli, we should observe that the N400 component for stimulus 2—which peaks about 400 ms after "leave" is presented—is somewhat higher than that for stimulus 1 and "go," while the N400 component for stimulus 3 and "dance" is higher still. Incongruity isn't a binary distinction but a graded one. The most vexing incongruities will tend to cause higher N400 peaks than those that do less to strain our expectations and confound our imaginations.

Whenever we process a humorous stimulus, the N400 offers a reliable marker of the folly of choosing path A over path B, or of favoring system 1 over system 2, with a spike in negative potential announcing the surprise at the heart of the joke. If ERP components were buses, the N400 would be among the most reliable, for while the amplitude of the deflection is sensitive to various factors, these typically have little effect on its arrival time. Nonetheless, those that affect the magnitude of the peak can tell us a lot about what influences our perceptions of incongruity. Naturally, the more surprising the stimulus, the bigger the peak, while any aspects that diminish its unpredictability also tend to reduce the magnitude of our response. Although much of

the work of comprehension is done retrospectively, as when we work backward from a punch line to its setup, a significant element is still predictive in nature. As we process one stimulus, we prepare for the next by anticipating the words and ideas that may connect that stimulus to this.

We say that one word *primes* another of a related form, sound, or meaning, so “went” primes both “come” and “go”; “pizza” primes “pizzazz,” “Italian,” “cheese,” and even “pitta”; and “dish” primes “food,” “plate,” and “wish.” Priming sets up our expectations for plausible continuations, and so creates a basis for surprise when those expectations are eventually dashed. Each potential outcome can be graded according to the cumulative priming it receives. Take, for example, the familiar phrase “weapons of mass [X],” which primes not just the value “destruction” for [X], its strongest prospect, but also “instruction,” “construction,” “description,” and “distraction.” Of the many values for [X] that are primed in this way, “distraction” is the prospect that sits at the sweet spot for innovation on the priming gradient, halfway between what is truly unexpected and what is utterly familiar.

Recency also plays a role here. Once primed, a word or an idea gains salience for a time, but it will not hold our attention for long unless it is primed repeatedly. Perhaps this is what comedians mean when they say that timing is everything: a comedian who gives away too much too often, and overprimes the incongruity to come, will likely reduce the magnitude of the audience’s surprise. Conversely, one who primes a key idea too soon, or waits too long to provide a vital clue, may present the audience with an incongruity that makes no sense at all. Somewhere between these two poles is a Goldilocks zone in which the audience is optimally primed to resolve an incongruity that still retains much of its power to surprise.

The optimal innovation hypothesis of Rachel Giora and her colleagues holds that incongruous stimuli work best when we prepare the ground for their arrival.²⁹ The largest N400 peaks mark out the most confounding stimuli, but a sizable peak will still suffice for humor if it shows the moderating effects of a familiar context. A clever one-liner shows how moderation can result in subtlety *and* surprise:

My wife caught me wearing her clothes, so I packed *her* things and left.

This joke hinges on the second use of the word *her*, which has previously been primed by its first use (“*her clothes*”) and by the word *wife*. This second use is a surprise, but priming has made it a familiar surprise and an optimal innovation. However, not every surprise is a humorous surprise, no matter how optimal or familiar it may seem. Consider the line, “The king was pregnant,” from the science fiction classic *The Left Hand of Darkness*. Ursula Le Guin’s novel imagines a branch of humanity for whom sex and gender are not genetically determined.³⁰ Rather, the people of Le Guin’s planet, Gethen, have no fixed sexuality and become male or female each time they enter their reproductive periods, called *kemmer*. It is when the king enters *kemmer* as a female that “his” royal pregnancy ensues.

So what makes the one-liner seem funny but not very profound, and Le Guin’s line seem profound but not funny? Both lines benefit from the moderating effect of context, making it possible to grasp the gender incongruity when it comes. Just as we might have expected the one-liner to read, “I packed *my* things and left,” Le Guin’s line would better align with our expectations if it had read, “The *queen* was pregnant.” But long before we ever reach this line in her novel, she has carefully constructed a context in which queens can be kings and vice versa. This is not the only reason her line is incongruous but not humorous: unlike the one-liner, her line is not a self-contained narrative that contains its own resolution, and it holds out very little opportunity for us to put ourselves—via our theory of mind—into the shoes of its royal protagonist. The N400 peak can tell us a lot about surprise but very little about how it makes us feel when it arrives.

We might say that Le Guin’s line and the one-liner generate different kinds of incongruity. The humor theorist John Morreall identifies at least three kinds, which differ principally in how one responds to each.³¹ The first, and the most interesting for us, is amusing incongruity, the kind that produces a feeling of levity and playful nonseriousness; the second is the incongruity of the jump scare in horror movies and is the kind that leaves us discombobulated and fearful; and the third is the incongruity of rational surprise, the kind that leads us to update our models of the world and admit that what once seemed silly or impossible is indeed possible. The one-liner produces an incongruity of the first kind, while Le Guin’s line produces an incongruity of the third kind. The infamous scene in the movie *The Crying*

Game, which I won't spoil here, delivers an incongruity of the second kind, although viewers have been known to laugh as well as gasp at it, and even to broaden their views of gender as a result. Morreall's three types are not mutually exclusive, and a blended response to incongruity remains a possibility.

MIND THE STEP

ERP analysis gives us a physiological marker of the sudden fall in our geometric joke space. But the N400 is a symptom of incongruity, not its cause, and we can no more give a machine a sense of humor by fiddling with its voltage levels than we can make someone laugh with electroshock therapy. Just as the N400 peak can be precisely plotted along the two axes of time and voltage, we must also identify the specific axes that define a catastrophe-theoretic joke space if they are to work for us computationally. Figure 6.4 reminds us of what the 3D cusp looks like.

Each path across the surface represents a different interpretation process, which I have denoted A and B. Although they start at the same point, the paths diverge during the temporal progression of a joke. The arrows indicate that time flows along the x -axis, while the divergence between A and B occurs primarily on the z -axis. Since jokes deliver their content sequentially, X is both the axis of time *and* information. As time progresses, more information becomes available to A and B, even if each uses

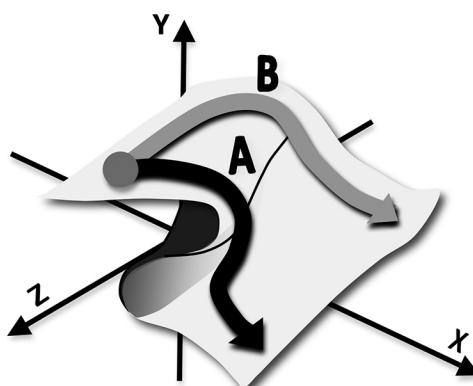


Figure 6.4

The cusp catastrophe in three dimensions permits divergent modes of traversal.

it differently. Eventually process A experiences a catastrophe on the y -axis that does not afflict process B. So what does the y -axis represent?

Humorous surprises are optimal when they achieve a balance of obviousness and novelty, as when, for instance, a conventional form is hijacked to deliver a fresh meaning. If the x -axis tracks a joke's information content over time, and the z -axis marks the obviousness of how this content is framed, then the y -axis tracks the expectedness of the result. In the case of process A, the dramatic tumble at the cusp marks the sudden realization that the most obvious interpretation has led us astray. In other words, using A to understand the situation produces a failure to match its specifics to an expected generalization. Or, in the data compression view championed by Jürgen Schmidhuber (in chapter 2), process A dramatically fails to achieve the expected compression of the stimulus because it has mislabeled it.

Consider the setup to our cross-dressing joke, “My wife caught me wearing her clothes,” and the two most obvious ways that this scenario can play out:

- A . . . so **she** packed **her** things and left. (*wife leaves*)
A' . . . so **I** packed **my** things and left. (*husband leaves*)

But the Frankenstein blend takes familiar elements from both of these outcomes:

- B . . . so **I** packed **her** things and left. (*husband + wife*)

B paints a far less predictable outcome than A or A' because our experience of life, language, and fiction tells us so. A piece of statistical machinery called a language model allows us to quantify these intuitions relative to a body of representative texts called a corpus.³² If we think of a language as the set of everything we can say in it, then a language model is a statistical approximation of the infinite resource that is our way with words. We cannot list all of the sentences in a language like English to say that this one is valid and that one is not, but a model can learn from a series of observations to assign a probability to every possible observation, seen or unseen. This corpus of observations should be large enough to inform the model not just about individual word frequencies but about the tacit knowledge that guides how we combine

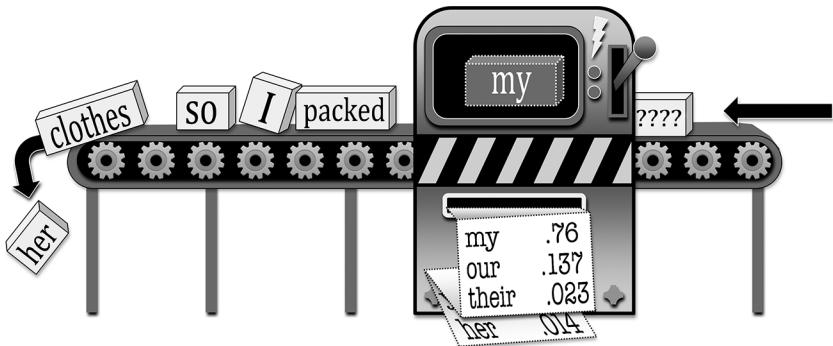


Figure 6.5

A language model allows a machine to make predictions about the likelihood of future word events on the basis of recent observations and past experiences.

words in context. If our language model is trained on vast amounts of Web text, its probabilities will tacitly incorporate a great deal of common sense.³³

As illustrated in figure 6.5, a language model is a predictive engine. For a given sequence of words, it can estimate the probability of that sequence being produced as part of its language, and it can also assign meaningful probabilities to the likelihood of future word usage on the basis of recent observations. That is, for a sequence of n words $w_0, w_1, w_2, \dots, w_{n-1}$, a language model will assign a probability to every possibility for word w_n . So given this sequence, “My wife caught me wearing her clothes so I packed . . . ,” the model gives us probabilities for the next word being *my*, *her*, and even *your*. If we denote the probability of observing the word w_0 in a language as $P(w_0)$, and the probability of observing w_1 as $P(w_1)$, then the probability of observing w_1 just after w_0 is $P(w_1|w_0)$, where $|$ is read “given that.”

More generally, $P(w_n | w_0, w_1, \dots, w_{n-1})$ is the probability that w_n is the next word that we see after observing w_0, w_1, \dots, w_{n-1} . To reflect alternate perspectives on the world, we can build and use different language models, so that $P_A(w_1|w_0)$ is the probability of w_1 following w_0 from perspective A, while $P_B(w_1|w_0)$ is the probability of w_1 following w_0 from perspective B.

As the information-bearing words of a joke arrive along the x -axis, we assume that the y -axis tracks the expectedness of what comes next. We can calculate this as a ratio of probabilities: the probability of what we actually

see next, over the probability of the model's best prediction for what might have come next. When its prediction is *my* and the next word is also *my*, the expectedness of this word from the perspective of interpretation process A will be calculated as:

$$\text{expectedness}_A = \frac{P_A(\text{"my"} | \text{"My wife caught me wearing her clothes so I packed"})}{P_A(\text{"my"} | \text{"My wife caught me wearing her clothes so I packed"})}$$

When the next word is *her*, process A calculates its expectedness as follows:

$$\text{expectedness}_A = \frac{P_A(\text{"her"} | \text{"My wife caught me wearing her clothes so I packed"})}{P_A(\text{"my"} | \text{"My wife caught me wearing her clothes so I packed"})}$$

In the first case, the expectedness score is 1, but in the second, it is much closer to 0. So, tracking expectedness on the *y*-axis leads us to an abrupt drop via the most obvious route, path A, when *my* is expected and *her* is observed. Conversely, for interpretation process B, the expectedness of *her* is calculated as follows:

$$\text{expectedness}_B = \frac{P_B(\text{"her"} | \text{"My wife caught me wearing her clothes so I packed"})}{P_B(\text{"my"} | \text{"My wife caught me wearing her clothes so I packed"})}$$

If process B anticipates that *her* is likely to be the next word, perhaps because its language model recalls the joke from a previous telling or because the model has been trained on a corpus containing a great many jokes, it will assign a higher probability to this event, and will not undergo a similar catastrophe to process A.

GOING WITH THE FLOW

Language models come in different shapes and sizes and can vary greatly in their internal complexity. Neural models, for instance, train artificial neural networks (ANNs) to capture the sequential flow of language from one word to the next. A recurrent architecture allows the outputs of the network to feed back into itself as inputs, so that the model is informed not just by newly arriving observations but by its memory of earlier decisions. Such networks comprise layers of many simple computing units that take their inputs from a lower layer (i.e., one closer to the input data) and feed

the results of their aggregation and thresholding operations to upper layers that successively transform them into a final output.

The multilayered connectivity of a typical neural network is illustrated in figure 6.6. Network architectures can vary in the number of layers, the number of units (or “neurons”), and the number of connections between the units in successive layers. In general, different layers serve different purposes. Depending on where they are placed in the network, layers may derive successive abstractions of the input data or perform successive elaborations of the output. The greater the number of units and connections between them, the more nuance the network can acquire from its training data, and the greater the scale of the data it can learn from and model.

The largest and most densely connected neural language models can be trained at Web scale to deliver truly impressive results. The generative

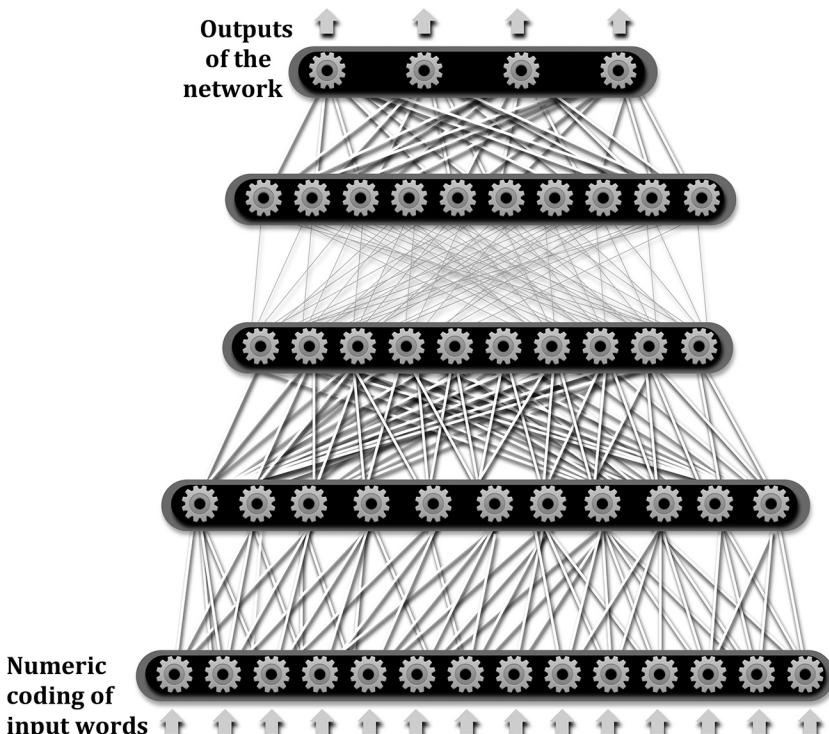


Figure 6.6

A general architecture for a densely connected, multilayered neural network.

pre-trained transformer (GPT) architecture, as developed by Open AI, has been used to train a language model with billions of parameters (corresponding to the weights of the connections between its units) on vast amounts of Web text.³⁴ This connectivity supports an attention mechanism that conditions the model's outputs on a large enough window of recent decisions that it is capable of producing stories from an input prompt that remain textually coherent as the story grows in size and detail. If you test-drive GPT's predictive powers for yourself online, you'll see that its stories are often original and fun, even though the model does not explicitly strive for novelty or incongruity.³⁵ Indeed, the opposite is true: by sampling the language model to suggest the most likely stream of observations that follow from the given prompt, the generator uses predictability, not surprise, as its guide. So, these models, while impressive as language savants, are ultimately just one part of a humor solution.

We might view a statistical language model as a codification of system 1, that is, as a mechanism trained to react to a stimulus in the most predictable ways. In contrast, system 2 doesn't so much react as analyze. It seeks to understand what it produces and concerns itself with the meanings of its choices. This is precisely what an old school approach to semantics is supposed to model. But if system 2 is to correct the mistakes of system 1, this semantic acumen must stand outside the natural flow of a language model, to evaluate, detect, and occasionally insist on a do-over, perhaps with a different model, in pursuit of a larger goal. When the goal is humor, it is the job of semantics to discern a meaningful opposition and a potential for overlapping frames or scripts in the outputs of the underlying generator.

This relationship is as adversarial as it sounds: humor emerges from a clash of perspectives, whether they happen to be organized as frames, scripts, or systems. Computational humor isn't any different, except that its clash of perspectives can also be a clash of levels, models, representations, and, ultimately, technologies. As we'll see next, nowhere is this clash easier to localize or to analyze than in a pun.

7 WIT HAPPENS: COMPUTATIONAL MODELS OF PUNNING AND WORDPLAY

SET PHASERS TO PUN

Puns are hard to avoid, yet they remain the only kind of joke for which we feel any need to apologize when we produce one by accident. Is this because we hold even deliberate puns in low regard, or because we associate them all, good and bad, with the schoolyard? Take this pun, a punning riddle to be precise, whose wit has not improved any since it first fell out of a Christmas cracker in the 1970s:

Q: How many ears does Mr. Spock have?

A: Three. The left ear, the right ear, and the final front ear.

You don't have to be a Trekkie to catch the echoing of form here, as Captain Kirk's opening voice-over has firmly imprinted the cliché "Space: The Final Frontier" into the minds of generations of kids who could take or leave this TV staple. That the pun instead attributes the echo to Mr. Spock has everything to do with how it corrupts its original form. Of all the show's cast members, Spock's unnatural ears make *him* the most natural target for this optimal innovation. Yet for all that, the pun is a groaner. We recognize the sound similarity and may even appreciate the writer's half-hearted efforts to make the echo meaningful in a Star Trek context, but the laughter it earns is as perfunctory as a mercy flush. Let's try to do better.

A pun is an optimal innovation that strikes a balance between the familiar and the unexpected. In spoken puns, the innovation arises in a word substitution that is motivated largely by sound similarity. The similarity

allows listeners to retrieve a more familiar stimulus and determine how the substitution affects its meaning. Of the following two innovations, each a potential slogan for a food delivery app, only the second is a pun, and it is all the more creative for that: “your *pizza* is my command” and “your *dish* is my command.” In each variation of “your *wish* is my command,” we can use the surrounding syntax as a guide, but only in the pun can we use the substitution itself as a basis for recovering the word that it replaces.

In this chapter, we look at puns from both ends—from the points of view of the producers *and* the consumers of this most disposable form of humor. We’ll see that it’s no accident that the kind of joke that children first delight in and that still slays in the schoolyard is also the kind for which computers have made the most progress. We begin by formalizing what we mean by an echo before we try to pin down the nature of the humorous distortions that twist these echoes into jokes. Knowing how to recognize puns will allow us to build, through a process of reverse-engineering, our own funhouse mirrors and algorithmic echo chambers. To start, we return to a topic we met in the previous chapter, the language model, to show how this statistical tool allows us to pull on the diverse threads of the World Wide Web to obtain the raw materials for our pun-spinning machines.

MODEL STUDENTS

How many words can you think of to sensibly complete the following phrase?

bad day at the _____

If this were a game show, you would most likely buzz in with the answer “office.” Our experience of the world is reflected in the words that come most quickly to mind, and “office” is undoubtedly the most frequent completion for this phrase. We can verify this intuition by looking to a resource called the Google n-grams database, which brings together a huge variety of text snippets from Google’s index of the Web.¹ We refer to any sequence of n contiguous words or punctuation symbols in a text as an n-gram, and Google’s database provides a comprehensive inventory of all 1-, 2-, 3-, 4-,

and 5-grams that occur at least forty times on the Web (at the time its inventory was compiled). The choice of $n=1, \dots, 5$ and the threshold of forty ensures that the n-gram inventory, though massive, is still manageable. A 5-gram like “bad day at the *office*” has a wide enough presence on the Web to suggest a high degree of linguistic conventionality, and perhaps even a degree of idiomaticity, too.

<i>Word 1</i>	<i>Word 2</i>	<i>Word 3</i>	<i>Word 4</i>	<i>Word 5</i>	<i>Count</i>
bad	day	at	the	office	5403
bad	day	at	the	beach	241
bad	day	at	the	tables	150
bad	day	at	the	track	120
bad	day	at	the	hash	108
bad	day	at	the	races	96
bad	day	at	the	dog	80
bad	day	at	the	roulette	61
bad	day	at	the	theatre	56
bad	day	at	the	plate	50
bad	day	at	the	range	49
bad	day	at	the	computer	42

Figure 7.1.

Google n-gram matches for “bad day at the __”

As shown in figure 7.1, lesser completions for “bad day at the _____” include “beach,” “tables,” and “track,” or words that modify these nouns, such as “dog” for “track” or “roulette” for “tables.” Let’s suppose that our listing is complete, so that every possible way of continuing the phrase “bad day at the __” is given, along with a count of all of its uses on the

Web. From these absolute counts, we can easily calculate the relative likelihood of picking one word over another and assign a probability to each. The probability of each continuation can be estimated relative to its 5-gram context, so:

$$P(\text{"office"} \mid \text{"bad day at the"}) = 5403/6456 = .837$$

Since the sum total of the 5-gram counts for “bad day at the ___” is 6,456, our estimate for the probability of the last word in the 5-gram being “office” is .837, while for “beach” it is .0373, “tables” .0232, “track” .0186, “hash” .0167, “races” .0149, “dog” .0124, “roulette” .0094, “theatre” .0087, “plate” .0077, “range” .0076, and for “computer” .0065. All of our estimates sum to 1.0 to define a full probability distribution.

In reality, however, our distribution is far from complete, since we have only considered the possibilities that satisfy Google’s threshold of forty Web uses. There are many other ways of completing each sequence that we will fail to count, just because they are too far out in the long tail to be listed as n-grams. Yet, even if we *could* count every single n-gram with a count as low as 1, we still couldn’t account for the propensity of future language users to creatively invent new ways of speaking, or to innovate optimally over what has gone before. If we want to build a predictive machine from an n-gram database that assigns a probability to any sequence of words, we will have to accommodate the unknown in our predictions too.

The bulk of the mathematics of language modeling, which we’ll gloss over here, concerns how best to accommodate the unseen into our models. Using the counts of Web n-grams leads us to overestimate the probability of attested sequences like “bad day at the track” and to assign no probability at all to unseen but plausible sequences like “bad day at the polls.” The probabilities of a complete distribution must always sum to 1, so we cannot simply assign a small nonzero probability to an unattested sequence without first shaving off a little bit of the probability mass assigned to all of the attested sequences. A variety of smoothing methods allow us to do this with more or less aplomb. As for the unattested sequences, since a pun occupies the middle ground between the familiar and the unexpected, we should be able to estimate the likelihood of a novel punning variation as a function of the probability of a more familiar, attested sequence. Look

again at the list of 5-gram completions for “bad day at the ___” and see if you can invent a completion that echoes *and* distorts one of those more conventional possibilities.

Although we can replace “tables” with “stables,” “races” with “laces,” and “track” with “trick” or “truck,” none of these substitutions seems in the least bit comical. Replacing “office” with “orifice” cuts closer to the funny bone, perhaps because this replacement word is somewhat rare and technical, with a hint of high-brow innuendo thrown in for good measure. Punning substitutions crucially rely on phonetic similarity, but the real test of a pun—or any other optimal innovation, really—is the new meaning it allows us to foist on a familiar form. There must be a reason for the substitution that builds on the sound similarity, so that the pun achieves semantic as well as phonetic cohesion. The choice of “orifice” isn’t funny in itself, but it can be *made* funny by motivating it with the appropriate joke setup:

Q: Why was the dentist in a bad mood?

A: He had a bad day at the orifice.

Even in this context, a naive language model will assign a zero probability to the occurrence of “orifice” after “bad day at the.” Yet it’s not as if the 5-gram “bad day at the orifice” is completely novel. It is a novel sequence built from familiar parts, and the probability we assign to it should reflect this piecewise familiarity. That is the job of a good back-off strategy. A back-off strategy allows an n-gram language model to estimate a probability for an unfamiliar n-gram by backing off to shorter attested sequences, by, for example, using 4-gram counts to estimate a 5-gram count.

The Google database assigns a Web count of 206 to the 4-gram “day at the orifice,” so even a simple back-off strategy can assign a nonzero probability to our punning 5-gram. Since our estimate for $P(\text{“orifice”} \mid \text{“day at the”})$ is already very small at .000345 (or 206/596,502), even a sophisticated strategy will assign a very low conditional probability to $P(\text{“orifice”} \mid \text{“bad day at the”})$. This is just as it should be: we want any creative or unexpected uses of language to be assigned low probabilities so that we can distinguish them from much more conventional uses. Let’s define our own punning

back-off strategy, called E for echo, that values any n-grams that might be optimal innovations, and scores them as such:

$$E(\text{"bad day at the orifice"}) = \text{sim}(\text{office:orifice}) \times P(\text{"bad day at the office"})$$

For puns, $\text{sim}(w:r)$ calculates a real number in the range $0 \dots 1$ that represents the phonetic similarity of a word w and its replacement r . While E returns a number between 0 and 1, E does *not* denote a probability and should not be treated as though it does. Nonetheless, E is still a reliable barometer of optimal innovation, since we can expect $E(\text{"bad day at the orifice"})$ to be significantly higher than $P(\text{"bad day at the orifice"})$, indicating that a targeted substitution has been made to a familiar n-gram on the basis of phonetic similarity. The difference in these two measures can be used to find likely puns in a text simply by seeking out n-grams with very low (or zero) probabilities for which a phonetic substitution yields a high E rating.

We can expect this difference to be high even if a pun is already present in the n-gram database. It turns out that Google assigns a count of 56 to “hard day at the orifice” for an unsmoothed probability estimate of $56/7092 = .0079$. If $\text{sim}(\text{office:orifice})$ is .7, we can calculate E as follows:

$$\begin{aligned} E(\text{"bad day at the orifice"}) &= .7 \times P(\text{"bad day at the office"}) \\ &= .7 \times .837 = .586 \end{aligned}$$

$$\begin{aligned} E(\text{"hard day at the orifice"}) &= .7 \times P(\text{"hard day at the office"}) \\ &= .7 \times .777 = .544 \end{aligned}$$

In each case, the E rating is far greater than the conditional probability of the pun, since $P(\text{"orifice"} | \text{"hard day at the"}) = .0079$ and $P(\text{"orifice"} | \text{"bad day at the"}) = 0$. So we can think of E as the creative echo and P as the signal that it magnifies and distorts. The degree of the magnification is the evidence a machine needs to spot a pun in a text or a developing conversation, but we still need a reliable measure of phonetic similarity to calculate E . A machine may well tackle this task directly by comparing its phonetic representation of the words w and r , or it can assume that the written form of the words is as good a guide to their pronunciation as any other, and simply compare both strings using a standard measure of *edit distance*.² For instance, we can turn “orifice” into “office” with two edits, by replacing the r with an f and

deleting the first *i*. The greater the edit distance between the written forms, the greater the phonetic distance is likely to be also. While spelling is not a reliable guide to pronunciation in English, a phonetic dictionary allows us to get around this problem. The CMU Pronouncing Dictionary has served the AI community for decades and provides entries for well over 100,000 word forms.³ Here are its phonetic transcriptions for “orifice” and “office”:

“orifice”: AO₁ R AH₀ F AH₀ S

“office”: AO₁ F AH₀ S

The entry for each is given as a sequence of symbols from the ARPAbet phoneme set.⁴ Each transcription indicates which phonemes should be stressed when the word is uttered in the normal way: no stress is indicated with 0, primary stress is indicated with 1, and secondary stress is indicated with 2. A machine is now on firmer ground when it concludes that “orifice” can be turned into “office” with two phonetic edits that preserve the stress pattern: just delete the R and the first AH₀.

Yet some edits should cost more than others. Swapping an unstressed for a stressed vowel, or a long for a short vowel, or a hard for a soft consonant, creates a more appreciable distance between two spoken forms and should be scored as such. So, while the distance between “job” (JH AA₁ B) and “jibe” (JH AY₁ B) or “job” and “jab” (JH AE₁ B) is just one phoneme in each case, “job” and “jab” sound more alike than “job” and “jibe.” In his 2003 PhD dissertation on puns, the humor researcher Christian Hempelmann defined a more nuanced model that assigns specific costs to the replacement of one phoneme by another.⁵ AI researchers can use his cost model as is, or train their own, since Hempelmann also provides an inventory of 1,182 pun/target pairs that serves as a training corpus from which machines can learn to assign an appropriate cost to every punning substitution.

This is exactly what three researchers at the University of Washington, Seattle—Aaron Jaech, Rik Koncel-Kedziorski, and Mari Ostendorf—set about doing.⁶ Their paper, “Phonological Pun-derstanding,” shows how a machine can detect puns with the one-two punch of a language model working together with a trained phonetic edit model. Their language model, which is trained on 230 million words of text, captures many of the linguistic intuitions of a native English speaker and so can be used

to test the viability of any sequences that are produced when a likely pun is replaced using the phonetic model. Those replacements, guided by a keen sense of phonetic similarity, aim to recover a pun’s original, unpunny form. For example, in this punning riddle, “How does Moses make his tea? Hebrews it,” their system swaps out “Hebrews” for “He brews,” since white space is cheap at the phonetic level. It then asks its language model to assign a probability to the resulting string. Unsurprisingly, the probability of “He brews it” in English is much higher than that of “Hebrews it,” so their model suggests that the recovered form is also much more natural as an English sentence than its optimally innovative derivation.

COWBOYS AND IDIOMS

We don’t need a probabilistic language model conditioned on Web n-grams to tell us that one sequence of words is more familiar than another. The raw counts of the n-grams often suffice on their own, especially if a pun plays on a cliché like “a job well done.” This is the key intuition behind a system called *Idiom Savant*, as implemented by my student Sam Doogan for the SemEval shared evaluation task on pun detection in 2017.⁷ SemEval is a recurring workshop with a bake-off format in which AI researchers build semantic systems that are evaluated on a common challenge and a shared data set. The 2017 pun task was organized by the humor researchers Tristan Miller, Christian Hempelmann, and Iryna Gurevych, and covered both homographic puns (where a pun word and its target have the same spelling, as in “I was a banker until I lost *interest*”) and heterographic puns (where each is different, as in “jab” versus “job”).⁸ We can make a similar homophonic/heterophonic distinction for spoken puns—just substitute pronunciation for spelling—but the SemEval task focused on written puns as found in online texts.

While heterographic puns typically stand out like a sore thumb, homographic puns hide in plain sight and can be distinguished only by their double meanings. It is the homographic (or, more precisely, homophonic) variety that we tend to produce accidentally—a side effect of priming—and feel a need to apologize for. Language models are of little use in detecting this kind of perfect pun because it blends so seamlessly into its context

of use. We must instead attune to secondary meanings that resonate with a larger theme of the utterance. For instance, “I lost interest,” is a banal way of saying, “I lost my reason to care,” but the resonance of financial “interest” with “banker” can turn this idiom into a pun. In heterographic or heterophonic puns, it is enough that the pun word and its target have different forms, and we don’t really need to know what either means, at least for detection purposes. But in homographic puns, these forms are one, and detection crucially relies on an ability to see the relevance of multiple senses in context. This requires a computational process called word sense disambiguation (WSD).⁹

So what two senses of “character” are activated by this pun from Nick Helm at 2011’s Edinburgh Festival Fringe: “I needed a password eight *characters* long so I picked Snow White and the Seven Dwarves”? What even counts as a word sense? Dictionaries have traditionally taken an actuarial view of the idea of word senses to assign a numeric order to the most stable arrangements of meaning that single words can be used to convey. Unsurprisingly, digital inventories of word senses tend to go with whatever works, especially since most of these lexical databases are themselves derived, in part or in whole, from conventional dictionaries. WordNet, for example, an electronic database of words and their senses, owes as much to the print dictionaries of old as it does to modern AI representation techniques.¹⁰

Conceived by the cognitive scientist George Miller to be a large-scale model of the mental lexicon—that is, of how our minds store, connect, and retrieve words—WordNet has become an important resource for computer scientists looking to build language applications using more than just a big list of words. WordNet gathers words with overlapping senses into sets of near-synonyms, called *synsets*, which it organizes into networks of meaningful relationships that can be traversed by a machine. It organizes its noun synsets hierarchically, allowing us to generalize by climbing up, from the synset *{pun, punning, wordplay, paronomasia}* to the synset *{fun, play, sport}*, say, and to specialize by climbing down, such as from this *fun* synset to *{comedy, clowning, funniness, drollery}*. The two senses of “character” activated by Helm’s pun are captured by the synsets *{character, grapheme, graph, graphic symbol}* and *{character, fictional character, fictitious character}*.

Nothing tests a WSD system like a perfect homographic pun. The 2017 pun task considered both homographic and heterographic puns from three perspectives: *detection*—Does a given utterance contain a pun? *location*—Which word in the utterance suggests the multiple senses of the pun? and *interpretation*—What are the salient WordNet senses of that pun word? So given the utterance, “Your wit is my command,” the ideal system should answer “yes” for question 1, “wit” for question 2, and the WordNet synsets *{wit, humor, humor, witticism, wittiness}* and *{wish, wishing, want}* for question 3, since these are the most likely senses of the word that is replaced (“*wish*”) and the one that replaces it (“*wit*”). Eleven systems from different countries and institutions competed in the 2017 bake-off and were evaluated on the same test data. These were compared not just to each other but to a range of naive baselines chosen by the organizers to determine whether the top-ranked systems really were all they were cracked up to be.

For pun detection, the baseline was a coin toss that randomly said “yes” 50 percent of the time. For pun location, three baselines were defined: the first picked a word at random from the utterance; the second chose the last word of each utterance, in line with our assumptions about punch line placement; and the third picked the word with the highest number of senses in WordNet. For pun interpretation, a random baseline chose two senses of the pun word at random from WordNet (for homographic puns only), and a most frequent baseline (for heterographic puns) chose the senses of the pun word and its target that occur most frequently in a sense-annotated text corpus. These truly are meant to be *base* lines insofar as we should expect any serious attempt at pun analysis to beat them hands down.

Heterographic and homographic (or heterophonic/homophonic) puns really are very different kinds of beast, at least from a computational perspective, so it’s not surprising that no one system dominated on both types for all three tasks. The Idiom Savant system described earlier, which uses the Google n-grams to situate a pun within a familiar phrase, came tops at the detection of heterographic puns, with a precision¹¹ of 0.87 (that is, 87 percent of its answers were correct) and a recall¹² of 0.7837 (that is, it returned 78.37 percent of the correct answers), to earn a combined F₁ score¹³ (the geometric mean of precision and recall) of 0.8439. Another system, Duluth, triumphed

when detecting homographic puns, earning itself an F_1 score of 0.8254.¹⁴ As these puns squeeze two senses into a single word, we can expect a WSD system to be torn between one sense and another. So Duluth ran four configurations of the same WSD system over each utterance and flagged a pun word when it observed a conflict among the senses assigned in different runs.

For the second task, the UWATERLOO system (from the university of the same name) was tops at locating heterographic pun words, with an F_1 score of 0.7964.¹⁵ It used a set of heuristics that weighs each word's place in an utterance, favoring those nearest the end, and the sum total of the information that each shares with every other word in the utterance. This sharing of information among words allows our sentences to cohere like structures made out of LEGO. For instance, in a sentence containing the word *carols*, the word *Christmas* is almost entirely redundant, and we can capture this useful intuition, which is key to the statistical analysis of meaning, with a measure called *pointwise mutual information* (PMI).¹⁶ The PMI of any two words is the log of the probability that both will occur in the same context, as divided by the product of their individual probabilities.

Idiom Savant exploits a similar intuition, but it does so for homographic puns only. Recall that Idiom Savant uses the relative counts of two n-grams—a familiar phrasing and a variant with a phonological substitution—to detect heterographic puns, so its detector is also a locator too, at least for this kind of pun. With an F_1 score of 0.6845, it falls short of UWATERLOO, although it is the only other system to best the naive last-word baseline for the location task. Idiom Savant is not able to parlay differences in n-gram counts into a means of homographic pun detection, so it reasons, much like UWATERLOO, that pun words blend into their contexts of use when they cohere with the words around them. To quantify this intuition, it uses Word2Vec, a geometric model of word meaning (of which we will see more in the next chapter), to calculate the similarity of each word to every other word in the utterance.¹⁷ It then selects the word with the highest total similarity to the rest of the utterance. This allows Idiom Savant to come out on top at homographic pun location, earning it an F_1 score of 0.6631 against UWATERLOO's 0.6523.

The outcomes become a good deal gloomier for task 3, pun interpretation, as no system does a creditable job of identifying the senses activated

by either kind of pun. It's one thing to use formal criteria to detect a joke and quite another to look past the words to see the meanings and intentions they convey. Determining the WordNet senses of a heterographic pun word and its target, or the dual senses of a single homographic pun, is only the beginning of the search for humor. The best result on homographic puns was earned by two of the bake-off's organizers, Tristan Miller and Iryna Gurevych, who achieved a precision of 0.1975 and an F_1 score of 0.1603.¹⁸ So, if given a punning utterance, their system has a one in five chance of correctly identifying the dual senses of the pun word within. Although Idiom Savant was top at pinpointing the senses of heterographic puns, its F_1 score of 0.0771 is notable only for being less risible than that of other systems, for which F_1 ranged from 0.0009 to 0.0303, and for beating (by a whisker) a naive baseline that picks the most frequent senses of the pun and its target.

Although sense identification falls short of actual interpretation, it is a crucial first step. Only when the specific senses of a pun word and its target have been identified can we begin to explore how they fit together. Just as importantly, only then can we look for incongruities in how they fail to fit together, to see if the friction between their conceptual frames is sufficient to spark a humorous effect. We have to start somewhere, but tasks that stop short of producing the meaning of a joke allow machines to take far too much of the comprehension process for granted. A machine can do well at these tasks without ever knowing what makes a pun funny, or why some get laughs and others earn groans, or why formal criteria for punning might also be satisfied by typos or innocent uses of ambiguous words. Using WordNet and the CMU pronunciation dictionary, our machines can generate scads of dubious puns that are good enough to pass muster with the best SemEval systems while still falling short of what it takes to make us humans take notice. To truly understand the hidden complexity of a humorous form like the pun, we must explore how a machine might successfully generate one in the first place.

BABY, I WAS BORN TO PUN

It makes sense to think that a system that can detect and appreciate jokes is well placed to invent them too. But most algorithms make too many

assumptions or take too many shortcuts to be so cleanly reversible in practice. Think back to how the Duluth detector used discrepancies between competing WSD systems to seek out homographic puns. A good WSD system uses all of the implicit cues it can find in an utterance to select the senses that best fit the evidence. This makes Duluth's job nicely reductive: given an utterance as input, return a single word as output. But it's not easy to see how we can reverse this: to give a single word as input and then obtain a whole utterance as output. When information is thrown away in one direction, it has to be imaginatively reinvented in the other. Yet while detectors like Duluth lack the wit to take a word like *flat* and invent a pun like "We charge a flat rate to fix your tires," they can still serve a useful role *post-generation*, as a means of filtering what other, more generative algorithms have cranked out.

Let's assume that our machine is given a pair of phonetically similar words and must invent a punning utterance around the substitution of one for the other. This is the formulation of the pun generation task preferred by He He, Nanyun Peng, and Percy Liang from Stanford University and the Information Sciences Institute.¹⁹ Consider how the word pair *died/dyed* is used as a pun in this utterance: "After I accidentally swallowed some food coloring I think I *dyed* a little inside." Before we tackle the challenge of inventing this utterance from scratch, let's peek ahead to see how a detection system like Idiom Savant might appreciate it as a pun. With *died* as our probe, the Google 5-grams suggest the contexts in figure 7.2.

Word 1	Word 2	Word 3	Word 4	Word 5	Count
I	die	a	little	inside	1268
I	died	a	little	inside	428
i	died	a	little	inside	68
I	have	a	little	inside	64
i	die	a	little	inside	59

Figure 7.2.

Google n-gram contexts for "I __ a little inside."

In contrast, the corresponding 5-gram that is explicitly given in the text, “I dyed a little inside,” cannot be found in the Google database and is assigned a count of 0. Nonetheless, we can estimate the strength of its punning echo as follows:

$$E(\text{“I dyed a little inside”}) = \text{sim}(\text{died:dyed}) \times P(\text{“I died a little inside”})$$

We can assume that $\text{sim}(\text{died:dyed}) = 1$, since these two words are homophones. By summing the Web counts of the second and third n-grams in figure 7.2 and dividing by the sum of all the n-gram counts, an upper bound on $P(\text{“I died a little inside”})$ of $(428 + 68)/1887 = 0.262$ can be estimated. We can likewise assume that the value of $P(\text{“I dyed a little inside”})$ is close to zero, as even a simple back-off strategy will assign it a small, nonzero value, such as 0.0001. The surprise of seeing the latter when we are expecting the former is key to the effect of the pun. In information theory, this shock of the new goes by the name “surprisal,”²⁰ and we express it as the negative log likelihood of the two diverging outcomes:

$$\text{surprisal}(\text{died/dyed}) = -\log \frac{P(\text{“I dyed a little inside”})}{P(\text{“I died a little inside”})}$$

Specifically, He, Peng, and Liang refer to this measure as the local surprisal value, since it localizes the surprise to a specific neighborhood of the larger utterance. In their formulation of the pun generation task, the machine knows that *died* will be replaced with *dyed* from the get-go, so $\text{surprisal}(\text{died/dyed})$ is a measure of the suitability of the context “I ____ a little inside” to frame the substitution as a pun. By anticipating how a system like Idiom Savant might later evaluate this choice of context, it uses surprisal value to help it make an optimal selection. So the real question becomes: How might the machine invent a context like this for itself?

But why invent when you can reuse and repurpose? In a large enough corpus of sentences, our machine is sure to find a few that it can nip and tuck to suit its needs. Generic sentences are to be preferred, as these are more easily pried from their surrounding contexts to produce a meaningfully self-contained utterance. In the Google 5-grams, we find the seeds of

many such sentences, from “man died of bird flu” (874) to “man died in police custody” (597). As a general rule, we want to avoid sentences with pronouns that point outside the utterance and prefer those with generic nouns such as “the parrot.” Using another heuristic that proved its value to the SemEval pun task, He, Peng, and Liang also rank candidate sentences by the position of the target word, since replacements near the end of a sentence are more likely to work as humorous punch lines. So when given the pair *hair* and *hare*, their system retrieves this sentence, among others: “The man stopped to get a *hair* cut.” Substituting *hair* with *hare* gives us the first draft of a pun with a reassuringly high surprisal value: “The man stopped to get a *hare* cut.”

This first cut is still some distance away from becoming a decent pun. After all, what does it even mean to get a “*hare* cut”? We will need to change something else in the sentence too, to motivate this *hair*:*hare* substitution. We might, for instance, swap “stopped” for “raced,” to yield a pun that taps into our shared knowledge of hares as fleet-footed animals: “The man *raced* to get a *hare* cut.” Alternatively, we might swap “man” for another kind of entity with an established relationship to hares, to yield the pun “The *greyhound* stopped to get a *hare* cut.” So that the pun sentence contains a pair of joke “handles,” to use the jargon of comedy writer Joe Toplyn, we must find another word in the sentence that we can meaningfully replace with one that has an even stronger semantic link to the pun word.

The SemEval pun task once again suggests the best tool for the job. Recall how a measure of semantic relatedness can be used to calculate the fit between a pun word and its surrounding text. For instance, the Waterloo system used pointwise mutual information as its measure of semantic fit, while Idiom Savant used a measure of word similarity to estimate this fit. He, Peng and Liang went a third way. They trained a language model over skip-grams²¹—n-grams that contain gaps, allowing short patterns to span distant parts of an utterance—so that their model can assign probabilities to gapped word sequences such as “greyhound ____ hare” or “raced ____ hare.” The secondary replacement task now seeks out another word in the utterance that can be replaced with one of the same general type (e.g., one

sentient entity for another or one transitive verb for another) that strongly predicts the pun word via the skip-gram language model.

When it works, the result is an optimal innovation that exhibits local surprise (or surprisal) and global familiarity, even if no generic corpus is likely to contain a line so perfectly accommodating of a pun about food dyes as “After I accidentally swallowed some food additive, I think I *died* a little inside.” Even so, He, Peng, and Liang report solid results for their approach when it is tested on a subset of the 1,099 word pairs collected for the SemEval pun task. As a source of ready-made sentences, they used Book-Corpus, a collection of text extracts from books, and then asked human judges to rate the funniness of the resulting puns—after retrieval, swapping and smoothing had been applied—via Amazon’s Mechanical Turk (AMT) crowdsourcing platform.²² In a few cases, their machine-generated efforts even outshone the human puns in the SemEval collection.

For instance, AMT judges gave an average funniness score of 2 (out of 5) to this pun for the word pair *better/butter*: “Well, gourmet did it, he thought, it’d *butter* be right,” while the human-crafted competition was a riddle that only managed an average of 1.5, “Why did the dairy churn? The less said, the *butter* . . .” Overall, the system does well against some firm baselines, although the pithier human effort usually comes out on top. Take the input *flour/flower*, for which the human effort of “Betty Crocker was a *flour* child” (average score 4.5) easily beats the machine’s enigmatic output, “Butter want to know who these two girls are, the members of the holy *flour*” (average score 1.5). A machine-crafted pun text can satisfy all the internal metrics you like, but it still has to make some kind of sense to be funny.

Despite that, statistical models do seem able to capture some valid cognitive insights into humor, not least because they allow us to reconcile different forces in the shaping of a joke. We’ve seen that local surprise is important if a joke is to jolt us out of the complacency of habitual language, but so is global coherence if this jolt is to carry real meaning. So, as top-down forces guide our choice of words and phrasing to best communicate the desired meaning, bottom-up forces nudge words into the familiar ruts and grooves of linguistic convention.

This inherent tension is explicitly captured in a statistical model of punning from the cognitive scientists Justine Kao, Roger Levy, and Noah Goodman.²³ In a generative sandwich of three layers, which is illustrated in figure 7.3, they model the word choices of a pun as the filling in the middle, shaped by the weight of meaning from above and linguistic concerns from below. For simplicity, we can assume that these concerns fall into two broad types: either an individual word is chosen to contribute to the meaning of the pun as a whole, or it follows naturally from the words that precede it, much as a language model would predict.

The secondary meaning of the pun, the conventional meaning over which the pun innovates, is denoted as m_2 , while h_2 denotes the target word that is implied but not given. For instance, in the dyed/died pun, h_1 is “dyed” (given) and h_2 is “died” (implied), while m_1 is *to dye* and m_2 is *to*

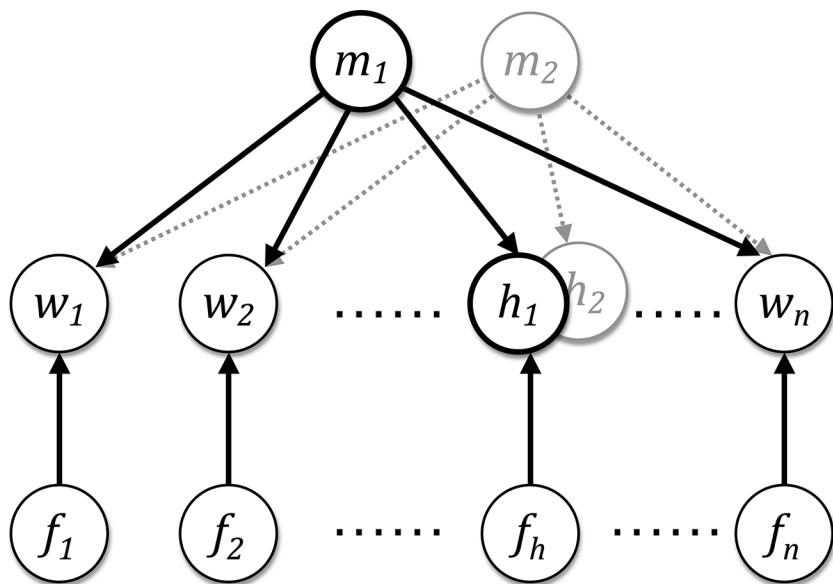


Figure 7.3

The top-down forces m_1 (humorous meaning) and m_2 (secondary meaning) and the bottom-up forces f_1, \dots, f_n (functional considerations) jointly influence the words w_1, \dots, w_n of a punning utterance. The phonetically similar pun word and its recoverable target are denoted h_1 and h_2 . Adapted from the Kao, Levy and Goodman's original figure.

die. The functional considerations f_1, \dots, f_n are simple, binary switches: when f_i is 0, the choice of w_i is dictated by the recent words that precede it, as suggested by a language model; but when f_i is 1, w_i is chosen for its semantic relatedness to the humorous meaning m_1 . However, w_i can't be just any word that happens to be related to m_1 . Rather, when f_i is 1, w_i is chosen to be h_1 , a word whose meaning is similar to m_1 but whose pronunciation is similar to h_2 , the word that *might* have been chosen had f_i been set to 0.

Kao, Levy, and Goodman identify two hallmarks of an effective pun: ambiguity and distinctiveness. A pun utterance $w_1 \dots w_n$ must enable two alternate meanings, m_1 and m_2 , and these must be distinctive enough to pose conflicting perspectives on the same underlying event. To measure the ambiguity of $w_1 \dots w_n$, they use the information-theoretic concept of entropy.²⁴ The greater the entropy of a message, the more information (or bits) we need to encode it without loss of meaning. In an ideal coding scheme, the number of bits needed to transmit a single meaning m of a word sequence $w_1 \dots w_n$ is given by its entropy, which we denote as $H(m)$:

$$H(m) = -P(m | w_1 \dots w_n) \times \log P(m | w_1 \dots w_n)$$

The ambiguity of a pun is a measure of the instability of its interpretation, which we can capture as the cumulative entropy of its different meanings $M = \{m_1, m_2\}$:

$$H(M) = -P(m_1 | w_1 \dots w_n) \times \log P(m_1 | w_1 \dots w_n) \\ - P(m_2 | w_1 \dots w_n) \times \log P(m_2 | w_1 \dots w_n)$$

As the secondary meaning m_2 becomes more viable, the cumulative entropy of M grows, and so too does the ambiguity of the interpretation. Nonetheless, m_1 and m_2 may not be distinctive enough as meanings to make for an effective pun. It is possible that any difference between m_1 and m_2 is just too subtle to register with a listener as a source of friction, as when, for example, $m_1 = \text{vegan}$ and $m_2 = \text{vegetarian}$. So we need to measure the meaning gap between m_1 and m_2 , since the bigger this is, the more dramatic the switch from m_1 to m_2 will be. Kao and her colleagues look at the words that can be used to convey m_1 and m_2 . By determining a probability distribution

over the words that are used to convey m_1 and another over those that can be used to communicate m_2 , they use a measure of divergence, the Kullback-Leibler score, to estimate the distinctiveness of the two distributions.²⁵ This score is a good proxy for the distinctiveness of the meanings m_1 and m_2 .

Armed with statistical measures of ambiguity and distinctiveness, we can now explore whether one is more conducive to the funniness of a pun than the other. Ambiguity quantifies our intuition that jokes are bistable texts that teeter on the cusp of two different interpretations, while distinctiveness allows us to estimate the potential of diverging interpretations to spark a humorous incongruity. Once validated on man-made puns, these measures may even provide a sound basis for determining whether a machine can predict the funniness of its own best efforts.

For their puns, Kao, Levy, and Goodman trawled the Pun of the Day website and collected nonpuns for the same pun words and their unspoken targets from other sources. A typical example is, “The magician was so mad he pulled his *hare* out,” an effort that is not beyond the ken of the retrieve-and-edit approach of He, Peng, and Liang. They collected 435 sentences in all, from perfect homophonic puns to imperfect almost-homophonic puns to indifferent nonpuns, before then recruiting judges on AMT to rate the funniness of each one on a seven-point scale. Their findings suggest that perfect and not-so-perfect puns do not differ significantly on measures of ambiguity or distinctiveness, or indeed, on human ratings of their funniness as puns. In stark contrast, puns and nonpuns *do* differ quite significantly on each of these measures, suggesting that a machine can reliably use ambiguity and distinctiveness to judge its own efforts.

Kao and her colleagues dug deeper to see if either of these measures can help us and our machines to predict the funniness of a potential pun. Specifically, they sought out correlations between each of these measures and the human ratings of funniness obtained from the AMT judges. Disappointingly, the ambiguity measure shows only a negligible correlation with funniness, even if it does prove its worth in separating puns from nonpuns. Fortunately, a much more robust correlation is observed between distinctiveness and funniness: puns that make it into the top quartile of distinctiveness are also rated, on average, as significantly funnier than those

that fall into the lower quartiles. So while statistical measures such as ambiguity, distinctiveness, and surprisal may not tell us how to invent a funny joke from scratch, at least not in any direct, do-this, then-that fashion, they can tell us how to filter whatever we do generate, however we manage to do it.

OLD SCHOOL

Kim Binsted's Joke Appreciation & Production Engine (JAPE), first graced newspaper headiness in the 1990s with an approach to the invention of punning riddles that put the form's most fervent users, school kids, in the driving seat of evaluation.²⁶ Working with Graeme Ritchie, her PhD supervisor, Binsted set out to implement a practical AI system for generating puns of much the same form and silliness as those found in Christmas crackers and children's joke books such as the *Crack-a-Joke Book*, *Super-Duper Jokes*, and *Super Silly Animal Jokes*. In contrast to a statistical language model, which defines a probability distribution over all of the possible sequences in a language, a symbolic model uses a smallish number of precise rules, typically handcrafted by the model's designer, to generate just that subset of a phenomenon for which success can confidently be predicted. For Binsted, this subset corresponded to the class of punning riddles—for example, “What do you get if you cross a sheep with a kangaroo? A woolly jumper!”—that creatively flesh out a relatively small set of structural and conceptual templates.

Binsted's JAPE has three well-defined structural layers: a schematic layer that understands the possible relationships between words, as defined in a dictionary or in a handcrafted representation; a description layer that can generate a short and semantically adequate description of a word; and a template layer that knows how to package words and descriptions into a surface form—typically a question—that users can appreciate as a punning riddle. What distinguishes the symbolic approach to AI is its treatment of domain knowledge. Rather than assume that the required knowledge is reflected in the co-occurrence patterns of words in a large body of text, most symbolic approaches set out to explicitly model this knowledge in a formal representation. What results is a semantic network, much like

the one for Starbucks that we drew in chapter 5. So JAPE formally links *kangaroo* to *jump*, *sheep* to *woolly*, *sweater* to *woolly*, and *jump* to *jumper*. It also links the ambiguous word *jumper* to both *sweater* and *one who jumps*. It can then use its schematic knowledge of riddles to chain these individual relationships together in just the right way, to generate the *sheep/kangaroo/jumper* pun we just saw.

For a pilot version of JAPE, denoted JAPE-1, Binsted hand-coded all of the word knowledge it would need to generate the puns in those wacky children’s books. JAPE-1 was a toy system, a working proof of concept that was difficult to scale in practice. But that’s where JAPE-2 came in: this would be no toy with a tiny lexicon, but would take most of its knowledge directly from WordNet and other off-the-shelf resources. JAPE-2 doesn’t need to be told that “jumper” is ambiguous or that its different senses, in different synsets, relate to things that jump and to clothes that keep us warm. JAPE-2 also uses psycholinguistic data to indicate the typical age of acquisition, concreteness, and imageability for the words in its puns. This allows JAPE-2 to shape language into the forms that most appeal to a child’s imagination.

As the pilot system, JAPE-1 was the first to be evaluated. Since this version was entirely handcrafted, Binsted took care to remove herself from the loop. Instead of augmenting the system herself, with the knowledge it would need to generate puns from the given seed words, she taught others how to add to JAPE’s store of relations and schemas. Adult judges in a second group were then tasked with the evaluation of any puns arising from the data entered by the first. When judges ranked 188 of JAPE-1’s efforts on a scale of 0 (not a joke) to 5 (a very funny joke), their mean rating was 1.5, with a mode of just 1 (corresponding to the label “A joke, but a pathetic one”). Nonetheless, the qualitative feedback from the judges proved useful to the development of JAPE-2, and Binsted was heartened to note that the judges had heard similar jokes before, so JAPE was on the right track.

To judge the outputs of JAPE-2, Binsted recruited the truest connoisseurs of the pun form, school children aged 8 to 11.²⁷ Care was taken to not expose the kids to unsuitable words and premises, and all puns were presented in both written and recorded forms to avoid any potential reading difficulties. In all, the kids were given four kinds of test stimulus:

sensible nonjokes in riddle form, such as, “What do you get when you cross a horse and a donkey? A mule”; absurd nonjokes in the same form, as in, “What do you get when you cross a dog with a suitcase? A light bulb”; punning riddles from JAPE-2 with the same shape, such as, “What do you get when you cross a grain with a murderer? A cereal killer”; and human-crafted puns, also in the form of a riddle, such as, “What do you get when you cross a computer with a lifeguard? A screen saver.”

A cohort of 122 kids served as raters on a test set of one hundred JAPE-2 jokes, sixty human jokes, twenty sensible nonjokes, and twenty absurd nonjokes. For each stimulus, the kids were asked to label it as a joke (1) or not (0), and scores for the same stimuli across different raters were averaged to produce a jokiness rating between 0 and 1. The children were also asked to rate the funniness of each effort on a scale from 1 (“not funny at all”) to 5 (“very funny”). Overall, JAPE-2’s efforts earned a mean jokiness rating of 0.6, compared to 0.81 for the human puns. In contrast, both kinds of nonjoke (sensible and absurd) scored in the region of 0.2.

For funniness, humans again outdid the machine, but not by a large margin. While JAPE-2’s puns garnered a mean funniness of 3.14 out of 5, the human efforts earned a mean funniness of just 3.57. Although this difference is statistically significant, these mean ratings place each kind of joke somewhere between “not sure” and “funny” on the funniness scale. So, while kids rated human puns as funnier and more joke-like, the machine’s outputs were still competitive. They were in the same league, if not always in the same class.

In fact, the joke that earned the highest funniness rating from the kids was not invented by a human but by JAPE: “What’s the difference between leaves and a car? One you brush and rake, the other you rush and brake.” Forget adult judges; these kids know what they like, and JAPE seems able to deliver it, at least some of the time. This partiality to puns prompted Binsted to speculate on the benefits of using computer-generated puns in educational software for children, especially when they are invited to partake in the generation process as active cocreators.

After all, part of the schoolyard allure of puns is the way they expose the norms of grown-up language to subversion and ridicule, allowing kids

to temporarily take control. A system that helped children with complex communication needs—arising from any number of physical or psychological causes—to invent puns of their own would allow these kids to share in the same cheeky delights. This is the motivation for STANDUP, the logical successor to Binsted’s JAPE designed by Ruli Manurung and Binsted’s original collaborators, Graeme Ritchie and Helen Pain.²⁸

STANDUP, which stands for “system to augment nonspeakers’ dialogue using puns,” offers a route to greater language confidence by turning learning into play. When children approach STANDUP’s interface, the overt goal is not to learn something but simply to share a joke. While puns are generated using the same schematic model and many of the same resources as JAPE-2, STANDUP also adds a picture dictionary and a speech synthesizer to boost its appeal among kids. Like JAPE, STANDUP takes care not to speak down to kids with words that are too formal, too abstract, or too technical, and it uses an extensive blacklist to avoid producing unsavory puns like this one: “What do you call a capable seed? An able semen.” Teachers were also asked to suggest their own words for the blacklist, perhaps based on their own experiences of awkward questions in the classroom.

STANDUP was evaluated in a pilot study of nine 8- to 13-year-old students with cerebral palsy at a special needs school in Scotland. While it is difficult to draw quantitative insights from so small a group of children who are not tracked in a longitudinal study, Manurung and his colleagues noted that the kids enjoyed using their software. Not only did they return to it unbidden, they eagerly shared with others the jokes they cocreated with it. The one child who complained about the quality of those puns even offered advice on how the system might be improved. While STANDUP and JAPE may succeed in fostering a new generation of computer-assisted comedians, they are also doing their bit for the future of heckling.

EXCESSIVE VOWEL MOVEMENTS

We apologize for puns more than for any other kind of joke, perhaps because no other kind is so easily produced by children or by accident. Yet even deliberate puns can present a valid a case for regret. Both kinds

spring from our search for words that best match the occasion, and since this search encourages us to intermingle superficial concerns about form and sound with deeper considerations of what it is that we really want to say, it's not surprising that even when we don't set out to seduce others with our word choices, we can still end up seducing ourselves. It is this strong emphasis on superficial issues of pronunciation and spelling that makes punning so alluring as a computational task. After all, AI's most trenchant critics have always argued that machines can really only manipulate form, not the meaning that it carries, so AI theorists necessarily conflate both levels.²⁹ However, since puns also conflate these levels, machines are free to focus on the generation of coherent word sequences while allowing meaning to largely take care of itself.

Ultimately, the distinction may be more philosophical than practical. Since we humans use meaning to guide our choice and ordering of words, any model that can reliably predict word sequences will implicitly capture much of their latent meaning too. It's not that modern AI is indifferent to the importance of meaning, but that it sees no point in replicating old school approaches to the representation of meaning in a formal symbolic language. Why trade a natural language for an artificial one when real language can speak for itself, in all its data-rich subtlety, through statistics? Still, for all the robust generalization power that language models and other data-driven approaches can buy us, we often yearn for the simple, top-down control of a symbolic system like JAPE, with its precise rules and schemas.

Yet there *is* a middle ground, in which we let the language data speak for themselves most of the time, and occasionally command those data into the global forms that we want them to take. As we go beyond the computational pun in the chapters to follow, it is in this sweet spot between top-down rules and bottom-up data that we shall try to position ourselves. This is a distinction that will prove to be especially relevant in chapter 9, when we look at irony, a mode of humor that requires us—and, someday, our machines—to bend words in one direction and meanings in another. However, before we do that, it will help if we first do the math.

8 PHYSICS ENVY: QUANTITATIVE APPROACHES TO HUMOR ANALYSIS

JUST DO THE MATH

Whatever else computers might be or do, no one doubts their prowess as number crunchers. Although it is the “else” that concerns us here, even this comes back to numbers in the end: a machine that sees the world as a math problem can still crunch anything at all that can first be converted into numbers. In fact, computers pull off this neat trick every time we use them. They convert between points of light and numeric pixels, between sounds and numeric frequencies, and between text and the number codes of the ASCII or UNICODE standards. So adept are they at performing this trick that only programmers ever catch them doing it.

How far can we get—or, indeed, how deep can we go—by treating humor as a math problem? When it comes to the possibility of translating between jokes and numbers, the jokes themselves are ambivalent at best. Consider a joke that is old enough to have spawned a number of intriguing variants. A newly convicted felon is lying in his prison bunk on the first night of his sentence when he hears other prisoners holler numbers from their cells. Someone yells 42 and everyone laughs. Another shouts 77 and more peals of laughter fill the night air. Puzzled, the new guy prods his cellmate for answers. “Well,” the jailbird explains, “we make our own entertainment here, but when it comes to jokes we’ve heard them all. To save time, we’ve given each joke its own number, and we just shout out the numbers of the ones we want to tell.” Keen to try this out for himself, the new guy yells 102 at the top of his lungs, but earns only an embarrassed

silence for his efforts. Minutes later, another convict hollers 102, but this time everyone laughs. Looking again to his cellmate for answers, he is advised that “it’s all a matter of *how* you tell ‘em.”

The numbers are merely indexical in this telling of the joke. They mean nothing in themselves, hence the incongruity of the punch line, but they do permit random access into the ordered list of jokes that every convict has already memorized. It takes a variant of the joke to suggest the possibility that these numbers might also be meaningful on their own terms. This version picks up where the first leaves off, with the new prisoner willing to try his luck again. “1023!” he shouts through the bars, to the raucous acclaim of the other convicts. Puzzled by the scale of their response, he turns to his cellmate, who is now in fits of laughter, and says. “That was a good one, I guess.” His cellmate replies, “It sure was. We ain’t heard *that one* before!”

How could anyone unpack the humorous potential of a number from its digits alone? It takes conviction, apparently, but it also helps if you live in a joke. A third variant of the joke is my personal favorite. The protagonists in this version are not prisoners but salty submariners, home on leave after months at sea. At a party for the crew and their wives, the men gather in the kitchen to drink vodka. They yell numbers between shots, each one eliciting louder peals of laughter than the last. When the wife of a young crew member expresses confusion—“What’s so funny?” she asks—the other wives tell her of the crew’s numeric code for retelling jokes at sea. At this, a drunken sailor hollers “96” in anticipation of approving laughter, but is met only with shocked looks and hushed silence. Finally, an elder submariner shakes his head and solemnly intones, “Not in front of the ladies.”¹

Jokes can make anything seem possible, even when they contradict themselves. That is one of the reasons we love them. Another is that we each bring our own baggage to a joke. When I hear the last two versions of the number joke, the geek in me can’t help but drag in the concept of Gödel numbering. We met the Austrian mathematician Kurt Gödel and his famous incompleteness proof in chapter 6, but let’s take a closer look at the building blocks of that proof. Since Gödel wanted to derive a mathematical proof *about* mathematical proofs, he first needed to turn the latter into the kind of thing that proofs normally talk about: numbers. Gödel

devised a way of mapping any mathematical statement—any sequence of formal symbols, really—onto a number. Once every proof, theorem, or conjecture can be assigned its own *Gödel number*, Gödel’s proof is free to speculate on the existence or otherwise of statements with a certain number within a given formal system.²

Gödel wasn’t the first mathematician to do this, although his approach was the most comprehensive. The German philosopher W. G. Leibniz had earlier sought to put the realm of argumentation on a mathematical footing by offering a means of converting arguments into numbers.³ In his scheme, every primitive idea is coded as a unique prime number, allowing composite ideas to be coded as the product of their component parts. Two arguments are compatible if the Leibniz number for one is cleanly divisible by the number for the other. In this way, Leibniz hoped to make the philosopher’s call to arms not “Let us argue!” but “Let us calculate!” The Leibniz scheme has some obvious limitations—How can it work for negated ideas or disjoint sets of ideas?—but he was on the right track. Gödel also used primes as the basis of his numbering scheme, taking care to code not just the variables in a statement but the operators that connect them, all in a position-sensitive fashion. Since statements about Gödel numbers have Gödel numbers too, he was then able to construct a corrosively self-referential statement to sit at the heart of his proof.

In principle, his numbering scheme can also be used to unpack a Gödel number. Formally, each statement is *encoded* as a number, while a number can be *decoded* as a statement. This complementary pairing of encoding and decoding operations is central to any attempt to arithmetize meaning and treat natural language as a math problem. It may be a stretch to imagine all of those prisoners applying Gödel numbering to their jokes, but I like to think the submariners might make a go of it. In practice, however, the old salts would be forced to exchange gigantic numbers with hundreds or even thousands of digits and would need to drink an awful lot of vodka. So when computer scientists map concepts and word meanings into the realm of mathematics, their encoding of choice is not a single number but a vector with many real-valued parts that marks out a point in a high-dimensional space.⁴

Vector spaces are fluid and continuous, and they permit the kind of operations that make little sense for our brittle words and symbols. So we use an encoder to map symbols into a vector space when we need to stretch, bend, or blend them in ways that are not formally defined in their own domains. We can then use a decoder to map the results out of this space of seemingly meaningless numbers and back into the world of intuitive symbols. Just think of the subsymbolic realm of the vector space as a libertine nightclub in the basement of a respectable hotel. What goes on below may be unthinkable above, but can have meaningful outcomes nonetheless. In the rest of this chapter, we explore how actions in this numerical basement can have predictable consequences at higher levels of linguistic meaning, allowing AI researchers to turn their number crunchers into effective joke crunchers too.

READY PLAYER TWO

The appeal of word games is not limited to prisoners and submariners, although the best ones do show a suspicious similarity to drinking games. They give us a social reason to explore the space of words and ideas and a scholarly excuse to drink to our successes and failures. If you like your games to be win-win, there are no losers in a game called Convergence, just minor setbacks on the road to shared meaning. As taught to the participants of a Dartmouth symposium by the digital humanities scholar Nick Montfort, the game also requires its players to celebrate with a special dance when this convergence of meaning is achieved. This dance is one more reason, if one were needed, to treat the whole affair as a drinking game.

A round of Convergence begins on the initiative of a single player, who has in mind a specific word. This player declares “One” to the group, prompting another to spontaneously reply “Two” once he or she has also settled on a word. At this, both players count to three and announce their words in unison. In the unlikely scenario that they each blurt out the same word, convergence has been achieved and the mortifying dance ensues. When the two words differ, the goal shifts to achieving convergence in the next round. However, the group now has a common focus for its word

choices. Subsequent players should seek to converge at the imaginary half-way point between the meanings of the two previous words, and make their next choices accordingly. Let's suppose that the first word choices are *alcohol* and *politics*. In the second round, two players suggest *party* and *schmooze*. In the third round, the choices are *cocktail* and *flirting*. The group finally converges in the fourth round once two players jointly guess *small talk*.

As a game of Convergence takes place in the real world, it also plays out in the realm of vector spaces. Although the players traverse their own personal vector spaces, we can assume—to the extent that each reflects many of the same cultural influences—that each player occupies a shared space. In round 1, the initial word pair marks out two positions in this high-dimensional space. Let's not worry for now about what each dimension means, or concern ourselves with the numeric value assigned to each. Let's just denote the two words as w_1 and w_2 , and denote their respective positions in the vector space as $\text{encoding}(w_1)$ and $\text{encoding}(w_2)$.

In the first round we can expect that $w_1 \neq w_2$, so $\text{encoding}(w_1) \neq \text{encoding}(w_2)$. In their scramble for convergence, two players in round 2 each propose a word that they believe is closest to the conceptual midpoint of w_1 and w_2 . But what is this word? Fortunately, in vector space, we can do things with vectors that we cannot easily do with words or ideas. We can, for instance, simply average both vectors, and assume that whatever the midpoint word might be, its vector encoding will be close to $\text{average}(\text{encoding}(w_1), \text{encoding}(w_2))$, or, in other words, the vector that we get by summing the corresponding dimensions of w_1 and w_2 and dividing by 2. To turn this new vector back into a word, we call on our decoder to arrive at $w_3 = \text{decoding}(\text{average}(\text{encoding}(w_1), \text{encoding}(w_2)))$.

If it were really this straightforward and deterministic, the group would always converge to w_3 in the second round of the game. That they do not, in most cases, is due to a number of important factors. First, the players aren't really traversing the same vector space. The metaphor of a vector space is merely a mathematical approximation of what is in each player's head. Second, they are not using exactly the same encoding and decoding procedures, for much the same reason. When two or more computers play against each other and use the same vector space, the game always ends in

the second round, since each machine will calculate the same w_3 . Yet when one computer plays against one or more humans, the game may go on for as long as the human-only variant. The fact that the game still converges to a satisfying resolution when a machine is in the loop shows that the machine's vector space need not be a perfect mirror of human knowledge. For a machine to interact with us on our terms, through natural language, it just needs its vector space to capture enough of our intuitions about words and meanings to converge toward a shared understanding in a reasonable number of steps.

We ensure the quality of the approximation by building our vector space from a large body of representative human texts. For instance, we can use the collected digital works of Project Gutenberg, (which contains almost 60,000 free ebooks).⁵ If we want our space to capture the implicit assumptions that underpin our use of words in jokes and other humorous texts, we must also add enough of these kinds of texts to the mix. Vector spaces do not directly capture the meaning of words, but work on the assumption that words show their meaning in how they are used with others. By encoding a word as a vector, we make no claims as to the meaning of the dimensions or the values assigned to them. It is enough that similar words sit cheek-by-jowl in the same space because the encoding process has distilled the different contexts in which they occur into similar values on the same dimensions.

HIDDEN FIGURES

Convergence is a game with a moving target that is pushed about the vector space by every incorrect guess. A related game, but with a fixed target, goes by the name of French Toast. In this variant, a volunteer (whom we will call the “arbiter”) picks a word or concept w_0 and keeps it in mind without disclosing it. As other players try to guess what w_0 might be, the arbiter signals whenever a guess moves the group closer to the target. It is the unique way in which the arbiter does this that gives the game its name. At first, the arbiter replies “French Toast” to any guess w_i that is more similar to *French Toast* than it is to the target w_0 . We can think of

“French Toast” as the arbiter’s default response. However, if a guess w_i is more similar to the target w_0 than it is to the default, the arbiter sets the default to w_i and offers this as a new response. Successful guesses narrow the gap between the default response and w_0 , and the game finally ends when guess $w_i = w_0$.

It’s fun to play arbiter and make wacky decisions that never arise in real life. Is a *bathtub* closer to *French toast* than to *world peace*? How about *love* and *beer*? It’s easy to imagine how this game plays out in a vector space because, like many other guessing games, it turns similarity into a matter of distance. The better the guess, the closer it takes us to our target. Vector spaces support a number of useful ways for assessing the distance between vectors and so, inversely, the similarity of the corresponding words and ideas. Since a vector marks a point in a geometric space, we can use standard geometry to measure the length of the line that connects the points marked out by two vectors w_i and w_0 . In 3D space, vectors mark out points with x , y , and z coordinates, and so the distance between two points is simply the square root of the sum of the squares of their differences along the x , y , and z axes, that is, $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$. In spaces with hundreds of dimensions, we extend the formula to sum the squares of the differences along all dimensions. This Euclidean or as-the-crow-flies distance is just the length of the shortest path between two ideas—such as *French toast* and *world peace*—in a vector space of ideas.

But a vector is more than a point. We can think of it instead as an arrow with a magnitude and a direction that points to certain coordinates from the origin (or zero point) of the space. Rather than measure the Euclidean distance between the coordinates that two vectors point to, we can measure the angle between their directions. A small angle tells us that they point at things that seem close together when viewed from the origin. Better still, we can calculate the cosine of this angle. The cosine of zero degrees is 1, the cosine of a right angle is 0, and the cosine of 180 degrees is -1 . A measure named *cosine similarity* gives us a normalized scale of similarity that ranges from -1.0 , when two vectors point in opposite directions, to $+1.0$, when they point in exactly the same direction at things that seem identical from the origin.⁶ This intuition is illustrated in the 2D projection in figure 8.1.

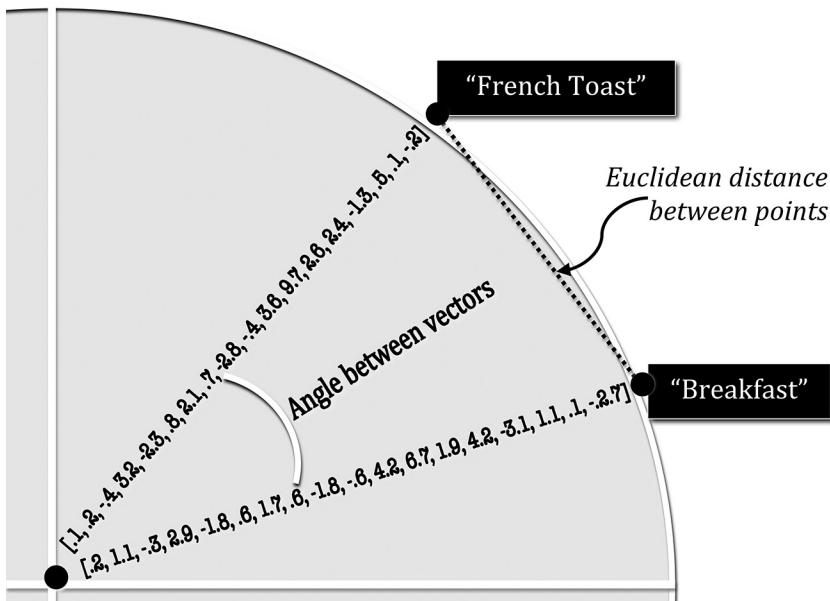


Figure 8.1

A 2D slice of an eighteen-dimension vector space. The angle between the vectors assigned to “Breakfast” and “French Toast” is a measure of how different they are.

A vector representation for words can be learned by an artificial neural network as part of its training for another task, such as the filling of word gaps in a window of surrounding text. We can train up our own word vectors—which are also called encodings or embeddings—on our own data, or we can use vectors that have been pretrained by others on a diverse body of linguistic data, such as a large corpus of Google news stories. One such resource, Word2Vec, has been trained in this way to map from words onto vectors of several hundred dimensions.⁷ Another source of pretrained vectors is GloVe (for global vectors), which also allows us to map from words to vectors and back again as the need arises.⁸ What matters in a vector is not the precise value of any specific dimension, but whether any two words that are close in meaning are encoded as vectors that are just as close geometrically.

The meaning of a word dictates how it should be used, but the converse is also true: how we use a word is a good indicator of its meaning. This intuition allows a technique known as latent semantic analysis (LSA)

to use word distribution as a proxy for meaning when constructing a vector space.⁹ For a given corpus, LSA populates a co-occurrence matrix by assigning a column c to each document and a row r to each word. At the intersection of every column c and row r , it places the relative count of word w_r in document d_c . To make the weightier words stand out from the rest, it doesn't use raw counts but TF.IDF (term frequency by inverse document frequency) scores, in which the number of times that w_r occurs in d_c is divided by its count in the corpus as a whole. As a result, words like *the* and *of* have much lower TF.IDF scores than meaty content words like *love* and *death*. The vector for a word w_r can now be read off as its row r in the occurrence matrix.

If we use this initial matrix as a basis for a vector space, then words will appear similar only to the extent that they have a similar distribution across all of the documents in the corpus. However, a matrix factorization method called singular value decomposition allows LSA to greatly reduce the dimensionality of the space while retaining, or indeed enhancing, the latent similarity between words implied by their distributions.¹⁰ So we can, for instance, build a 300-dimension space from the texts of the British National Corpus (BNC), which has a wide range of topics and styles, or one of equal dimensionality from a corpus made up entirely of jokes.

Professional comedians from Jerry Seinfeld to Joe Toplyn have emphasized the metaphorical power of an “angle” on a topic. Whether they use Word2Vec, GloVe, or LSA, vector spaces give a precise mathematical form to this metaphor. Since we can calculate the angle between a topic and a perspective when both are encoded as vectors, it is meaningful to think of an angle on a topic as the vector encoding of the perspective it represents. The narrower the angle, the more normative the perspective. Conversely, the more divergent the angle and its perspective, the greater the potential for showing the topic in a new and incongruous light.

Homogeneity fosters simplicity. Vector spaces allow us to aggregate multiple—perhaps many—words and ideas into a single vector by averaging their individual encodings. They also allow us to quantify the similarity of the coded possibilities, regardless of whether a vector encodes a primitive idea or a whole book of words. Nothing prevents us from averaging our

vectors for adjectives, nouns, and verbs to construct a semantic composite of the whole, or from encoding an entire text and decoding the nearest emotion in the space¹¹ (so Kafka’s *Metamorphosis* is “sad”), or encoding a feeling like sadness and then decoding the vector of the nearest color or object in the space, to get *gray* or *coffin* or some other semantically apt choice. As in that libertine swingers’ club in the basement of a respectable, rule-bound hotel, conventional categories simply do not apply at the vector-space level of meaning. We impose such categories from outside, as they are needed, and not from within.

SAFETY IN NUMBERS

Homogeneity also fosters anonymity, so it is rarely meaningful to examine a single vector in isolation. The dimensions of a vector space do not need names, and even when they have them, we may have no rubric for making sense of their values. Yet we can still use these dimensions as the features of a problem representation and treat each vector as a specific set of feature values. After all, it scarcely matters to a machine whether these features and values have human-readable names or not.

A feature is any element of a problem representation that can help a machine to make a decision. The process by which system builders choose and encode the features of relevance is called *feature engineering*. AI engineers have traditionally taken a rather parsimonious approach to feature selection, since the complexity of a system can grow with the number of features it must consider. At least that *was* the approach, when CPUs were slower and memories were smaller. In a modern machine-learning context, we can give a system as many features as we know how to represent and let the machine learn for itself, as supervised by annotated data, which ones are most discriminating for a task. What features of a person’s life and past would you use to decide whether someone deserves a teaching job, a low-interest loan, or a kidney transplant? You might be surprised by the features a machine learns to value, and even more surprised by the efficacy of its choices.

If only the AI of verbal humor, or of language more generally, were more like physics and less like stamp collecting, to use the put-down made

famous by Ernest Rutherford in his brutal assessment of the softer sciences. Physicists manage to derive powerful generalizations about the physical world from a rather small set of universal features. It is in the nature of such laws to work every time and in every context in precisely the same way, but humor—or, indeed, common sense—is rarely so systematic. As a past employer of mine would drill into the new hires at his AI start-up, common sense is not physics and AI has nothing at all to gain from physics envy. The Keatsian dogma that the truth is always beautiful simply does not hold, and it is a fool’s errand to seek to reduce complex, context-sensitive phenomena such as humor to beautifully elegant formulas like $E = mc^2$. My old boss, boorish but brilliant, had us throw the kitchen sink at every problem, dirty dishes and all, to offer up as many potentially useful features as possible.

This philosophy is nicely illustrated by a paper on joke detection from Rada Mihalcea, Carlo Strapparava, and Stephen Pulman, in which they consider how to distinguish short jokes from any old lines of text.¹² They compiled a corpus of short jokes with a feed line and punch line structure, such as, “Don’t drink and drive. You might hit a bump and spill your drink,” and recruited judges to suggest unfunny variants with the same lead-in. So when they were given “Don’t drink and drive. You might hit a bump and . . .” the judges suggested “get a flat tire,” “hit your head,” and “have an accident.” From the continuations elicited for each feed line, the researchers picked the three they deemed to be the most natural and the least incongruous. They then set about trying to identify the feature set that would reliably allow a machine to distinguish the original punch line from its three dull companions.

The first feature they explored was the similarity of each lead-in to its different continuations. Imagine a vector space into which we can map any joke. We expect the ones that most resemble each other, like our trio of number jokes, to map into nearby points in the space. When we break a joke in two, as was done in this situation, we might expect both halves of the joke to nestle cheek-by-jowl in the space. Mihalcea, Strapparava, and Pulman used LSA to build a pair of vector spaces, one from the British National Corpus (BNC) and one from a corpus of sixteen thousand jokes

that they harvested from the Web. We might hope that a generic corpus like the BNC contains enough tacit knowledge to do the job, but let's see.

The joke completions create an appropriate incongruity, while the serious ones are just appropriate. The question is whether the incongruity of the jokes affects the similarity of lead-in and completion in some discernible way. It turns out that in the BNC-derived space, the angles between the vectors for the lead-ins and the vectors for their corresponding completions are no guide at all to the humor of the combinations. The appropriateness of the incongruity appears to drown out the incongruity itself, and so, by picking completions on the basis of vector space similarity to the lead-in, a machine picks the original funny completion only 20 percent of the time. Contrast that with the 25 percent success rate of a purely random baseline. It seems that the BNC space is just too generic, or just too square, to "get" the joke.

Fortunately, the LSA space built from a corpus of jokes does a much better job. Similarity is relative in a vector space, so our sense of similarity can change as we move from one space into another. In a vector space built from jokes, geometric similarity judgments will implicitly incorporate a sense of how jokes tend to view the world, and this is exactly what Mihalcea, Strapparava, and Pulman show. When they use the LSA joke space as a basis for judging the similarity of each lead-in and completion, the most similar choice is also the funniest choice 75 percent of the time.

Other features that these authors explored turn out to be the icing on the cake. Lexical ambiguity, for instance, allows a joke to veer off the path of conventional interpretation into incongruous territory, so comics tend to prefer commonplace words with several senses and conflicting meanings. By favoring continuations with the most senses per word, on average, a machine might exploit this tendency to sift jokes from nonjokes. When used in isolation, however, the strategy is little better than random choice and picks the witty completion just a third of the time.

Comedians also tend to choose words for how they sound, since the rhythms of language create expectations in the minds of an audience that jokes can subvert or reinforce. Consider how Michael Caine defended his role in *Jaws: The Revenge*: "I have never seen it, but by all accounts it is terrible. However, I have seen the house that it built, and it is terrific." The

rhyme may be subtle, but for me, Caine’s choice of *terrific* as a counterpoint to *terrible* makes his quip all the funnier. So, all things being equal, a machine should prefer the completions that exhibit a greater degree of alliteration and rhyme with the lead-in texts. However, things are rarely equal, and this weak strategy also works just a little better than random choice.

A third feature, which looks at the overlap in domains between the lead-in and its continuations, is just as weak. Using a dictionary that maps from words to apt domain labels, such as *sport* for “tennis” or “racquet,” a vector of domain labels is constructed for every lead-in and continuation, so that the BNC vector space can be used to measure their cosine similarity. This feature also beats random choice, but not by very much. Nonetheless, a combination of weak features can often reap dividends in the end. So, when Mihalcea, Strapparava, and Pulman roll everything—strong and weak features alike—into a single, composite model, they achieve an overall accuracy of 85 percent when distinguishing the jokes from the nonjokes.

Physics envy encourages us to look for a small set of powerful generalizations that seem to capture the essence of a phenomenon, but empirical results show the value of combining as many weak generalizations as we can as long as they have an effect on the bottom line. So what other features can we throw at the problem of humor detection? The answer seems to be as many as we can. In the course of his PhD research with Paolo Rosso, Antonio Reyes drilled deeper into the kinds of features that worked for Mihalcea, Strapparava, and Pulman.¹³ Consider the ambiguity of not being able to reliably predict what comes next, as when a joke that starts with a warning about drunk driving ends with a sly tip for drunks. This is also the kind of ambiguity that best fits with the notion of the *perplexity* of a language model.

Some uses of a language will always seem more perplexing than others, and we can capture this intuition in a statistical model. The perplexity of a language model on a sample of text is the inverse probability of the sample as calculated using the model. More formally, the perplexity of a model for a sequence S of n words is the n th-root of 1 over the probability that the model assigns to S in the language.¹⁴

Reyes trained an English-language model on the Google n-grams and applied it to the sixteen thousand exemplars in the joke corpus of Mihalcea

and her colleagues. These are all positive examples of wit, so Reyes built a corpus of negative exemplars by mining sixteen thousand random sentences of comparable length from the Web. Calculating the perplexity of the English model on each sample lends support to the intuition that funny texts are less predictable than unfunny ones; the model’s perplexity for the positive sample is 0.07 and just 0.06 for the negative sample. The difference is not very large, but it suggests a possible role for perplexity in detecting humor.

We can drill deeper still to understand the specific kinds of ambiguity that add to the perplexity of humorous language. Take syntax. We find it useful to tag the words of an utterance with their parts of speech (POS), but words can be slippery, with ambiguous words evoking multiple parts of speech at once. In the quip, “The coffee tastes like dirt because it was *ground* yesterday,” the word *ground* evokes two senses simultaneously, each able to play a different syntactic role in the utterance. There is the verb sense of *ground*, specifically the past participle of *grind*, and its sense as a noun when it denotes the earth beneath our feet. Then there is the kind of ambiguity that changes the whole structure of an utterance, rather like the wireframe image of a cube that changes orientation as we shift our gaze from one part to another. So in the sentence, “The dog is ready to eat,” the dog can be understood either as a hungry consumer or as an exotic consumable.

Reyes counted the numbers of POS tags that are potentially considered when a sentence is processed. The perfect pun *ground* contributes two tags to this count. He estimated this count for tags of different kinds—specifically, noun, verb, and adjective—in every sentence of the positive and negative samples that he earlier used to compare perplexity scores. He normalized his counts by dividing each by the mean sentence length (in words). For the positive sample of sixteen thousand one-liners and the negative sample of sixteen thousand random Web sentences, the mean normalized counts for nouns (0.08 versus 0.07) and for verbs (0.11 versus 0.08) are higher for one-liners. Only the adjective counts register no difference across samples.

To quantify structural ambiguity in the samples, Reyes dissected the outputs of a parser to count the number of clauses in each utterance and

summed the mean number of verb relations and noun relations per clause. He averaged this measure of per sentence syntactic complexity for all of the sentences in the positive and the negative samples and found that the jokes have much lower complexity scores than their dull-witted counterparts (0.99 versus 2.02). It seems that humorous uses of language pull off a neat trick: they bubble and fizz with all kinds of ambiguity, yet bottle this energy in ways that seem beguilingly simple and straightforward.

To turn his humor recognizer into a full-fledged irony detector, Reyes added a slew of extra features. His pointedness feature scores the use of capitalized words and attention-grabbing punctuation, while his counterfactual feature is sensitive to words such as *however* and *yet* that contrast two different viewpoints. His temporal compression feature uses words such as *stop*, *suddenly*, and *now* to capture rapid changes in viewpoint, while temporal imbalance is set to high when a text contains contrasting verb tenses, such as past and present. Diverging noun senses are likewise reflected in a feature named contextual imbalance. This uses a semantic distance metric to assess the mutual dissimilarity of the ideas in a text.

Reyes also used various word-level features that can be directly pulled from a psycholinguistic dictionary.¹⁵ These associate words with scores for psychological arousal (e.g., *enjoy*=2.2, *avoid*=1.7), sentiment (e.g., *love*=3.0, *hate*=1.2), and imageability (e.g., *snow*=3.0, *doubt*=1.0). To this witch’s brew he added a variety of sequential features. Character-level n-grams capture substrings like “LOL” that a machine can learn to associate with a speaker’s ironic intent, while skip-grams—word-level n-grams that tolerate gaps in the sequence—allow for more formulaic templates, such as “a fine <gap> indeed.” Integrating sentiment with skip-grams allows machines to acquire generic templates that match against sentiment labels (such as <*positive*> for “brave” or <*negative*> for “dire”) rather than actual words. Overall, Reyes’s choice of features is motivated as much by the availability of expedient resources as by any cognitive or linguistic insight into the workings of irony. If we disavow physics envy, we can also jettison the philosophy that less is more.

There are no slam dunks in his volley of features, but that is all the more reason to lob so many at the task. Each does its job in unspectacular

fashion, contributing to the success of the whole at detecting irony. Reyes trained his irony classifier to use all of his features and tested its accuracy on data sets that were gathered from Twitter using the hashtags #irony, #education, #politics, and #humor. All but two of his features were found to be more prevalent in the #irony data set than others. The two exceptions—pointedness and temporal imbalance—are more commonly found in the #humor data set. He trained and tested his classifier using balanced (50:50) and unbalanced (30:70) splits of the ironic and nonironic data, the latter to reflect the reality that irony is less frequent than nonirony in everyday usage. His classifier generally achieves 70 to 75 percent accuracy on balanced splits of the data and 75 to 80 percent accuracy on unbalanced splits. In the next chapter, we will see that irony detection can achieve greater accuracy still when a machine is allowed to invent the most discriminating features for itself.

ARE YOU NOT ENTERTAINED?

Many good jokes are ruined in the telling. Sadly, it's easier to kill a joke with bad timing and poor phrasing than it is to create new ones from whole cloth. However, when it comes to teaching machines to discern jokes from nonjokes, the negative exemplars can be just as important as the positive ones. Mihalcea, Strapparava, and Pulman asked the crowd to suggest dull alternatives for each one-liner but picked out the least witty suggestions themselves. In contrast, Reyes looked to the Web to retrieve a new set of sentences with the same distribution of line lengths as the originals. Chance dictates that a randomly chosen line won't be funny, but neither will it have an obvious twin in the positive data set. So what if we could do better than either of these approaches and ensure that each humorous example has an unfunny twin from which it deviates in the smallest meaningful way?

The key is to draw the positive examples from a source of optimal innovations that values fidelity to a familiar norm. Consider the *Onion*, a satirical news site that uses journalistic tropes to package its humorous takes on politics, culture, and history. Its faux headlines speak to newsworthy

concerns, using phrasing that would not be out of place on the front pages of *USA Today* or the *New York Times*. Satirical headlines like “Tiger Woods to Return to Sex” or “BP Ready to Resume Oil Spilling” look close enough to the real thing to produce a double take when we spot the unexpected twist in the tail. It is this optimally innovative take on shared expectations and topical issues that drew Bob West and Eric Horvitz to the *Onion* as a trove of positive examples that imply their own negatives.¹⁶ Each unfunny negative lurks just beneath the joke’s surface, typically just an edit or two away. The ingenuity of their approach lies chiefly in how they coax the negatives to the surface by making a serious game out of the process of killing the joke.

Their game, called *unfun.me* (which is also the name of the website where it can be played), involves two kinds of players. The first edits satirical *Onion* headlines to generate serious alternatives that might grace the pages of a real newspaper, and the second assesses headlines for satirical intent. Players of the first kind are scored on the lightness of their edits and the believability of the edited headlines, while players of the second kind are scored for their ability to tell faux headlines from the real thing, whether satirical ones from the *Onion* or real headlines from an actual paper. The players do not interact directly, but their respective reward functions incentivize them to work against each other, like the goalie and striker of opposing soccer teams. The players of the first kind maximize their scores by removing the satire from a faux headline with minimal changes, and players of the second kind maximize theirs by refusing to let a badly edited headline pass for the real thing. Laziness on either side is always punished with a diminished score.

As an example, let’s suppose that player 1 is asked to edit the satirical headline “God Diagnosed with Bipolar Disorder.” To defuse the humor, this player might swap in “Trump” for “God” in a low-cost, one-for-one edit. Although the resulting headline could well be true, player 2 is inclined to see it as satirical, since “Trump” is a popular target of online humor. However, if player 1 instead replaces “God” with “Kanye West,” this two-for-one edit incurs a higher cost that may lower the score, but player 2 is now more likely to rate the modified text as a real headline.

West and Horvitz populate their game with 9,159 satirical *Onion* headlines and 9,000 serious headlines. No attempt is made to conceal the joke status of the *Onion* headlines that are given to player 1 to *unfun*, but when a modified headline is later passed to player 2 to rate for seriousness, it is paired with an unmodified headline that is either a known positive from the *Onion* or a known negative from an actual newspaper. To avoid bias, this pairing is presented in random order, before player 2 is asked to quantify a belief in the seriousness of each. Since the machine can access the ground truth of only an unmodified headline, it is on this basis alone that player 2 is scored. So to score well, player 2 must assign strong ratings to the seriousness of real headlines and much lower ratings to that of satirical ones. Because player 2 does not know which text in a pair carries the ground truth and which is merely another's attempt at a humor lobotomy, neither can they know for sure which headline rating will be used to calculate the reward. This gives player 2 a strong incentive for honesty when assessing both items in each pair.

As in all games with a purpose (GWAPs), the reward structure of *unfun.me* incentivizes players to create a useful by-product from their gameplay, in this case the matching of humorous headlines to the norms on which they innovate.¹⁷ So for every positive exemplar from the *Onion*, a machine can pick the best negatives for itself by looking to see how often, and how optimally, each persuades the players of the game that it is the genuine article. The scoring functions not only ensure the acceptability of every negative, they bias the machine to reward those with short edit distances that localize the humorous effect to a specific word or phrase. This allows West and Horvitz to derive empirical insights about the humor of satirical headlines and to speculate on the most practical strategies for creating new ones. If we think of the process of “unfunning” a satirical headline as the reverse of how a comedian—or a writer for the *Onion*—might inject humor into a real one, then knowing the best places to edit a text, or the most common kinds of edit to make, can help us in our efforts to turn our theories of humor into a practical science.

By the time West and Horvitz published their 2019 paper, their *unfun.me* game had elicited seriousness rankings for 2,794 edited *Onion* headlines

from a cadre of 546 volunteers. Most edits (more than 60 percent) take the form of substitutions (e.g., “sex” for “golf”); deletions without replacement are less common (a bit more than 30 percent), and additions without deletion are even less so (below 10 percent). Most changes are made to the nouns or noun phrases in a headline. Significantly fewer edits are made to verbs, even accounting for the fact that verb phrases are less frequent than noun phrases in headlines, and even fewer edits are made to adjectives and prepositions. It’s telling that more edits target the final phrase in a headline than any other part, no doubt to capitalize on, and subsequently dash, the expectations that are nurtured in the preceding parts. This is the case for headlines of all sizes, whether a headline offers three, four, five, or six different phrases for players to edit. It seems that players treat satirical headlines as one-liners whose stings should be placed where effective punch lines are conventionally placed in jokes: in the tail section.

As an optimal innovation, a satirical headline has a pointed meaning all its own, but each exploits many of the same expectations that drive the serious headlines too. These familiar (if unfunny) expectations are accessible to informed readers of the *Onion*, who can recognize, for example, that Tiger’s return to “sex” is just a humorous analogue to his earlier return to “golf” after a long-running sex scandal. At the heart of the relationship between the unfunny reality (Athlete returns to sport) and our satirical view of it (Tiger returns to the front page for the wrong reasons) we find an analogy, such as *Golfer:Golf::Tiger:Sex* or *Big_Oil:Drilling::BP:Spilling* (to use the preferred format of the Scholastic Aptitude Test). For Horvitz and West, these analogies must be faulty, since they make odd bedfellows of ideas that oppose rather than support each other. False analogy is just one of myriad logical mechanisms that have been proposed to explain how jokes contrive to squeeze overlapping but incompatible scripts into the same overloaded text.¹⁸ Nonetheless, news satirists clearly seem to favor this one mechanism over its logical brethren.

What makes these analogies false, or at least faulty, are the striking oppositions that repel the ideas they equate. Sex is a private activity and golf a public one. Sex is subject to many taboos, but golf isn’t subject to nearly enough. The *Onion*’s poke at Woods evokes the private versus public

and sex versus innocence oppositions, and more besides. Humorous opposition is special. It draws attention to itself so that it, rather than the ideas that give rise to it, is the focus of the story.¹⁹ Bad analogies are easy to make, but it takes a special kind of friction to make their fault lines seem funny.

Humorous analogies must still work as analogies if they are to convey a useful truth, so it is preferable to think of them as faulty rather than false. The *Onion's* Tiger Woods headline usefully reminds us that Tiger's private life is a public issue, making his sex life a spectator sport. West and Horvitz label the opposition of sex in the *Onion's* original headline to golf in its unfunned version as *obscene* versus *nonobscene*, to capture the sense that Tiger's virtuosity as a golfer is now tainted by his inability to separate his private and public personae. In fact, they labeled all 254 single-substitution cases in which a satirical *Onion* headline was successfully unfunned by swapping just one word for another. They assigned the generic high versus low opposition to 68 percent of cases, which is consistent with the idea that satire is used to pull the high-and-mighty down to our level. They then assigned more finely grained labels to these cases: 15 percent are *sublime* versus *mundane*, 14 percent are *successful* versus *failed*, 13 percent are *authority* versus *nonauthority*, 10 percent are *sophisticated* versus *simple*, 6 percent are *human* versus *nonhuman*, 5 percent are *human* versus *animal*, 5 percent are *new* versus *old*, 4 percent are *rich* versus *poor*, 3 percent are *religious* versus *unreligious*, and just 1 percent are *sentient* versus *nonsentient*.

The *Onion's* headlines rarely fit just a single opposition type. When it declares that "Fort Knox Receives \$85 from Cash4Gold," the contrast between *Fort Knox* and *Cash4Gold* suggests the opposition *authority* versus *nonauthority*, yet the contrast between Fort Knox and \$85 also suggests *rich* versus *poor*. The players of unfun.me can target different oppositions in such cases. While one replaces *Cash4Gold* with *Fed*, another replaces \$85 with *\$85 million*. The edits in each case are rather revealing about how satire works. Perhaps it is the need to say so much in so few words, but West and Horvitz suggest that satirical changes tend to foreground the negative in a situation. Conversely, many unfun.me edits reassert the primacy of the positive, since satire often achieves a critical edge by leading with a negative that asks us to imagine a more positive alternative. In this vein, the

Onion's headlines often paint pictures dripping with bad faith and stupidity, and readers must infer the brighter hues of the palimpsest underneath. So, when it announces "New Delicious Species Discovered," readers don't need to infer human rapacity—it is writ large in the headline—but they do need to contemplate why it should be otherwise.

We can expect a language model that is trained on a corpus of real headlines to register its version of surprise when we feed it faux headlines from the *Onion*. If we train a bidirectional model forward and backward, it will be equally adept at registering surprise whether the departure from the norm occurs near the front (as in "New Delicious Species Discovered") or at the end (as in "BP to Resume Oil Spilling"). But what makes writing for the *Onion* such a challenging task is that it turns surprise into the norm. Its satirists must surprise us with every headline.

Surprises are low-probability events, so it's hard to deliver the unexpected on a schedule. This creates obvious tensions for comedy producers that tie themselves to a single language model. Just as we cannot tickle ourselves, producers with just one model cannot surprise themselves. If we sample a language model trained on real headlines to create satirical ones of our own, we can expect to see meaningful surprises only about as often as we see them in print. Conversely, if we sample a model derived from the *Onion*, satirical outcomes will be frequent, but they will not be surprising. Even the wittiest *Onion* headline is surprising only relative to the expectations created by real news. To predictably create surprises often, and to appreciate them as surprises every time, we need to use two distinct models: one for a producer to sample so as to create events of the right form, and another to quantify the unexpectedness of these events from the consumer's perspective.²⁰

The dual model allows the producer to distinguish between the choices that the consumer expects and those that are most likely to surprise the consumer. We can visualize the distinction using the 3D cusp catastrophe of chapter 6, to see in figure 8.2 how the consumer and producer models interact across abstract space and real time.

This cusp is formed by interpolating the probability distributions of the producer and consumer models. Since each encodes expectations from

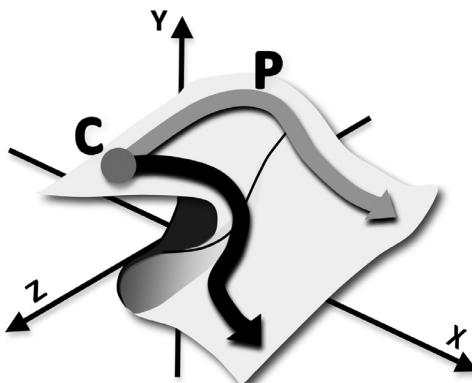


Figure 8.2

The cusp catastrophe, revisited. C and P denote the Consumer and Producer models.

different data sets, a low-impact choice for the producer can achieve a dramatic impact on a consumer. Since only the producer has visibility over the whole surface, it can make choices that deliver the greatest surprise to a consumer. Think of these models as defining competing probability distributions over the space of scripts, so, for example, a producer is inclined to see *sex* in a golfing context where consumers are disposed to see *sport*.

HOLD THE ONIONS

To explore different kinds of humor we need to tap into different sources of data. The *New Yorker* has published cartoons for almost ninety years, and while the pictures alone are often amusing in their own right, it's the captions underneath—often whimsical, sometimes cerebral, occasionally puzzling—that elevate these images into memorable jokes. Under the stewardship of its famed cartoon editor, Bob Mankoff, the magazine launched a weekly contest to entice readers to submit their own captions for some of its most striking images.²¹ Luckily for AI, the contest's popularity has produced a trove of data for researchers, who can now analyze the collected submissions of different runs with different images.

Consider the scene in figure 8.3: a salesman pitches a rather unusual car to a couple in need of persuasion. The couple is skeptical, to say the



Figure 8.3

Cartoon by Tom Cheney, *The New Yorker*, October 10, 2014. Used with permission. Licensed from Cartoonbank.com (Image ID: TCB-141085).

least, because this car has legs—four big, hairy ones—where it should have wheels. Its grille is a mouth filled with sharp metal teeth, and it has hungry, panther-like eyes where its headlights should be. The cartoonist, Tom Cheney, opted to caption his image with a reassuring line—“Relax! It just smells the other car on you”—but what would *you* have written? When the *New Yorker* based a run of its weekly contest around Cheney’s cartoon, these reader submissions were shortlisted by the judges²²:

What's it going to take to get you in this car today?

It runs entirely on legs.

Just don't tailgate during mating season.

It's only been driven once.

He even cleans up his road kill.

The spare leg is in the trunk.

Comfortably eats six.

She runs like a dream I once had.

Not wanting this trove of data to go to waste, Mankoff teamed up with Eric Horvitz and Dafna Shahaf of Microsoft Research to see what

computational analysis can tell us about what makes some captions appear funnier than others.²³ In particular, they hoped to give machines the same discrimination abilities, and broadly the same sense of humor, as the *New Yorker's* own panel of cartoon judges when it comes to ranking candidates for each image. This would allow a suitably trained machine to express the same preferences for a given image as a human expert or pick out the wittiest caption for any given cartoon, regardless of the image it adorns. More formally, our machine should tend to prefer the same candidates as human judges when given a pairwise choice of captions for each cartoon, and also tend to make the same selection as humans when given two captions from different cartoons.

Unlike headlines from the *Onion*, which can often stand alone as one-liners, the image to which a caption relates is an important source of its humor. Shorn of the image of a car with sharp, pointy teeth, the caption “Comfortably eats six” is a dull enigma that won’t tickle many funny bones. But ask yourself what must seem like a dumb question: Can a machine predict the relative wittiness of a caption just by analyzing the text alone, without looking at the image? You might be surprised by the answer, which is why computer scientists like to ask this kind of question first.

In fact, it’s a question predicated on some shrewd assumptions. The magazine’s readers can be expected to know a thing or two about what makes a caption work, and we can assume that most submissions for each run of the contest are sensibly grounded in the image. When Shahaf, Horvitz, and Mankoff pushed the captions for each run through a clustering algorithm, they found clear patterns of how this grounding is thematically realized. While the magazine received more than six thousand entries for its car-with-legs image, almost eleven hundred played on the idea of a hybrid car, three hundred or so referenced roadkill, and more than two hundred took a bestial view of the car. To the extent that a cartoon illustrates themes that are reflected in the words of its caption, we can expect its textual features to contribute more than its visual ones. Indeed, when it comes to pun captions such as “Comfortably eats|seats six” and “Relax! It just smells the other car|cat on you,” almost everything the machine needs is encoded in the caption. This is a big assumption, but let’s just go with it.

Shahaf, Horvitz, and Mankoff needed to establish a set of ground truths for the relative wittiness of different captions before they could demonstrate the value of language-only analysis, so they collected relative judgments of caption quality from Amazon’s Mechanical Turk (AMT) platform. Raters were paid a small sum to rank a mix of captions for the same image, from most to least funny, and from the totality of these orderings, a set of pairwise judgments was aggregated. When a caption pair was judged by five or more raters and 80 percent or more of them agreed on which of the two captions was the funnier, this pair with its aggregate judgment was considered a ground truth for training purposes. So, for example, if four out of five raters prefer “It runs entirely on legs” to “The spare leg is in the trunk,” then it is taken as a ground truth that the former is funnier than the latter. In this way, a machine can be trained to appreciate captions using the most reliable examples. In all, 754 caption pairs (just a third overall) reached the 80 percent agreement level.

To bring these details into focus, the team threw a range of linguistic features into the mix. Recall that the higher the probability assigned to a word sequence by a language model, then the lower its perplexity is estimated to be. Using a corpus of Web texts to train four n-gram models ($n=1, 2, 3$, and 4), Shahaf and her colleagues calculated the perplexity of the captions in each model. To the extent that surprise is a good predictor of humor, we should expect the wittiest caption in each pair to also have a higher perplexity. However, when perplexity is measured using the 1-gram, 3-gram, and 4-gram language models, the favored captions show a small but statistically significant preference for the least surprising forms. It seems that clichés like “It runs like a dream” and “What’s it going to take to get you in this car today?” work well as captions when their banality is a counterpoint to the image’s inherent incongruity. A bias toward familiarity is also evident when the reading difficulty of each caption is assessed with a standard metric and when perplexity is instead measured using a language model that is trained on part-of-speech tags.

In a similar vein, the most positive caption in each pair—in terms of the surface sentiment of its words—also tends to be ranked as the funnier of the two. Just as when a caption offers a familiar counterpart to an

unusual image, the positivity of a naive cliché can strike a humorously discordant note when the image calls for something as incongruous as itself. At first blush, “It runs like a dream I once had” exudes a positive sentiment, but this positivity curdles when it is paired with the unheimlich image of an absurdly bestial car. Likewise, the naiveté of the salesman who asks, “What’s it going to take . . . ?” also shows positivity where none is warranted, and this too is a source of humorous incongruity. The image and the text of a good cartoon each plays to different strengths: images allow us to make the absurd seem possible, while a familiar phrasing can make this possibility look ridiculous. For the *New Yorker* data set, Shahaf, Horvitz, and Mankoff show that the sentiment of a caption is a statistically significant indicator of wit even when we are blind to the elements of the image that elevate this simple linguistic sentiment into a joke.

Shahaf, Horvitz, and Mankoff trained a classifier for the pairwise judgment task to predict the caption in each pair that humans like more. The classifier uses a raft of surface features, the most effective of which are perplexity (for word sequences *and* part-of-speech sequences), readability, and sentiment. Since the machine has access to the crowdsourced “ground truth” for a large set of caption pairs, 80 percent of this set is used to train the classifier and the remaining 20 percent is used to test it. The held-out data do not overlap with the training data, and the captions come from different cartoons. Averaging its performance over different partitions of the data, the classifier has an accuracy of 69 percent when each pair refers to the same image. All things considered, this is quite impressive. The classifier never gets the joke, but it turns out to be a decent guesser when picking the ones that tickle *us* the most.

To train a classifier to judge the relative merits of captions for different images, we may need some new features, and we definitely need some new ground truths, so the team went back to the crowdsourcing well and presented each AMT rater with a mix of captions from a variety of *New Yorker* cartoons. They used the same criteria for sieving pairwise judgments from the sea of user rankings, to elicit a set of 4,184 robust ground truths as to which caption in each pair is the funnier. This step is as odd as it sounds: raters were given captions without cartoons and then asked to rank them

by humorlessness without knowing what they described. In a fascinating twist, while these judges were denied a peek at the associated images, the classifier itself was given a metatext of descriptive features for each picture.

In line with a two-script view of jokes, these metafeatures are partitioned into two sets, the normative (e.g., sales-related features for the *car-with-legs* cartoon) and the incongruous (e.g., features we don't expect to find in a car setting, such as sharp teeth and panther legs). Shahaf, Horvitz, and Mankoff used a vector-space model of word meaning to quantify the similarity of each word in a caption to the metafeatures in the two sets, and then reduced the results to a single absolute measure of affinity to either the more incongruous or the more normative aspects of the image. It turns out, to nobody's surprise, that the similarity of a caption to the image it describes is indeed a statistically significant predictor of its humor.

For the cross-cartoon comparison task, readability and sentiment scores are no longer statistically significant discriminators of humor, but perplexity (for word sequences *and* part-of-speech sequences) continues to be a telling feature. So too is the presence of recognizable names and brands in a caption, such as "Trump" and "IKEA." Overall, their new classifier achieves 64 percent accuracy when predicting the preferences of human judges for one caption over another in pairs taken from different cartoons. While it's not entirely correct to describe these preferences as humor judgments, since the judges who decide the ground truth for each pair can hardly be said to get the joke, the classifier does capture some of the aesthetic judgments that cause humans to prefer one caption over another. We need more than a pinch of salt to digest the machine's view of the best caption of any cartoon, but we can see how its views might help human judges to short-list the best ones.

DIM SUMS

Even when it yields compelling results for empirical tasks, it's tempting to dismiss AI's number-crunching of jokes as just an advanced form of numerology, one that brings us no closer to the spirit of humor than the jokes that opened this chapter. It is often said that to analyze a joke is to

kill a joke, and our mathematical analysis scarcely takes a holistic view of the patient. Rather like a veterinarian who diagnoses cows by first throwing them into a meat grinder, it cannot see the whole for its parts. Nonetheless, AI reveals that there are statistical regularities to jokes that we, as consumers, often fail to consciously register. The best producers can access these regularities as a form of comedic insight, which some may even pass on to others.

When Mike Cook and I wrote our book on Twitterbots, Bob Mankoff was kind enough to provide feedback on what our bots were doing and how they might do it better.²⁴ Bob has a shared interest in bots (he is the cofounder of Botnik Studios), and as a long-time cartoonist for the *New Yorker*, he has the comedic wisdom to know when a line is too long or too short, when it drops its payload in the wrong place, or when it hides its light under a bushel of verbiage. He critiqued the output of our bots and gave us valuable nip-and-tuck advice for future efforts. Perhaps this is where AI technology will have its greatest impact on humor: as a partner that looks over our shoulder, to offer advice about what does and doesn't work.

We will want this AI partner to be a straight shooter, so that it's not afraid to have or share its own opinions, or to really tell it as it is. But straight talking is not always plain speaking, or plain sailing for that matter, especially when a partner is chosen for its sense of humor. As we'll see next, a sense of irony equips us with ways of communicating the truth that are as effective as they are indirect. It's all a question of being able to separate form from content and feeling from function, to send propositional content in one direction and propositional attitude in another. Not only do we need our AI partners to appreciate our own attempts at irony, we must give them the tools to be just as playful when delivering their truths to us.

9 TAKING EXCEPTION: COMPUTATIONAL TREATMENTS OF SARCASM AND IRONY

SHAKEN, NOT STIRRED

Who doesn't like a cold, fizzy beverage on a hot day or thrill with anticipation at the distinctive *tsssst* of the can as it is opened? Language, like our beverages, can come in carbonated and uncarbonated forms. We use figurative devices like metaphor and irony to add effervescence to our words, so that descriptions that would otherwise fall flat might instead sparkle with vivid images and fresh ideas. Although we can make our words too effervescent and overcarbonate them with too many possibilities, we sometimes want an innocent-looking remark to bubble and fizz with layers of meaning that challenge others to discern our true intent. The writer Michael Chabon has memorably captured this unsettling ambiguity of irony in his novel *The Yiddish Policeman's Union*.¹ His hero, a divorced detective named Landsman, reports to a chief who is also his ex-wife. So when Landsman adopts a conciliatory tone and praises his ex, Chabon tells us that she "accepts the compliment as if it's a can of soda that she suspects him of having shaken."

Chabon's simile highlights an important point of disagreement between irony and metaphor. While the latter adds an alluring sparkle to a turn of phrase, irony adds the unspoken threat of a surreptitious shake, forcing us to decide whether to accept an offering at face value or reject it outright because humiliation may lurk within. The more subtle the irony, the more confounding the choice. If you've seen the film *Amadeus*, you will likely remember a scene in which the composer Salieri, proud but insecure, solicits the young Mozart's opinion on his latest opera.² The young man's words

seem to effervesce with approval, yet they conceal a scathing appraisal of his rival's abilities: "I never knew that music like that was possible." Suspecting a slight, Salieri humbly acknowledges the apparent compliment with an obligatory "You flatter me," for which Mozart again musters ironic ambiguity: "Oh no! One hears such sounds and what can one say, but—Salieri!" What can Salieri do but smile, politely but skeptically, unsure of quite what to believe?

What is fascinating about Mozart's responses is that, in his mind at least, both are perfectly true: Salieri's music *does* stretch the bounds of possibility, if not in a good way, and it does distinguish its creator with its singular arrangement of sounds. Ironic assertions like these work at two levels simultaneously: the level of propositions—of what exactly is asserted—and the level of attitude, specifically, the speaker's attitude to what is asserted. Sarcasm, a crass kind of irony, works at the propositional level, insofar as the speaker does not truly believe what is being asserted. More subtle variants work at the level of attitude: the speaker believes, or mostly believes, what is asserted by the utterance but presents a misleading impression of the emotions that accompany it. The most subtle form of all can confound the speaker and listener alike; well-crafted irony can provide so much plausible deniability to a speaker—"Ironic? Me?"—that they themselves may be unsure of their true disposition with regard to the underlying proposition.

A cynic might describe Mozart's replies as manipulative and deceitful, since he secretly scoffs at music and a musician he appears to praise. But a more generous soul might see those same replies as charmingly diplomatic, insofar as Mozart avoids giving offense in a delicate situation that expects certain social niceties to be observed. In fact, Mozart's responses are both of these things at once: sincere *and* insincere, truthful *and* deceptive. Irony gives him a way to be true to his real feelings about the music while being sensitive to the feelings of its creator. If you question the wisdom of granting computers a sense of irony—and why wouldn't you, given that our machines are already sullen and intractable enough as it is—this may be just the use case to change your mind. For who wouldn't want their machines to be more sensitive to their feelings and expectations, even if it means placing themselves at the end of an assembly line of teasing and gentle mockery?

The insincerity of sarcasm and irony can operate at the level of a proposition or our attitude to it, or at both levels together.³ Since we humans find it difficult to navigate the murky crawl space between the two, you might think that machines should find it nigh-on impossible. Yet despite the obvious difficulties, cognitive science makes the algorithmic modeling of human attitudes a cornerstone of its discipline. The cognitive scientist Steven Pinker summarizes the field's attitude to computation, as well as its computation of attitudes, as follows: "To put it crudely (and critics notwithstanding), beliefs are a kind of information, thinking a kind of computation, and motivation a kind of feedback and control."⁴ And what are irony and sarcasm but carefully controlled forms of linguistic feedback, toward a state of affairs that motivates certain emotional attitudes in the observer?

But what is really being controlled is the presentation of our emotional state to others, in ways that allow us to convey profound disappointment while remaining in full control of our feelings. While machines that deliberately lie are a danger to their users and to society, the insincerity of irony and sarcasm isn't intended to be an effective concealer of propositional attitudes. Rather, the communicative value of this insincerity ultimately resides in its failure to convince. It conveys our belief that things ought to be better than they are, and our disappointment that they are not. It recasts a negative emotion as a positive, but allows some negativity to leak out. Ironic speakers are poker players who *want* their bluffs to be called.⁵ As we'll see, they even use a variety of *tells* to help others to see through them.

The situation is complicated by the variety of ironies that we tend to perceive not just in language but in the world itself.⁶ Even everyday situations can be ironic—in what is called *situational irony*—if they appear to be undermined and made laughable by their own inherent contradictions. A no-smoking sign in the lobby of a big tobacco firm may be an example of health and safety laws in action, but it is also strikingly ironic. Verbal irony uses words to criticize a failure of reasonable expectations, as when a woman rebuffs a married man in a nightclub with the line "Does your wife know you're single?," but situational irony marries happenstance to action to show the world laughing at itself, or at us, as when a real tornado destroys a drive-in theater during a screening of the film *Twister*.

The form of irony most favored by scriptwriters, dramatic irony, highlights the contradiction between what characters know of themselves and what audiences know of them. In the 1968 movie *Planet of the Apes*, Charlton Heston's astronaut crash-lands on an alien planet where talking apes are the dominant species, but the audience, if not Heston, knows that all is not as it seems. Why else do the apes speak perfect English? A related form is the ironic plot twist, a turn of events that upends the expected outcome of a narrative to deliver a blow that is as apt as it is incongruous. In the 1982 film *Blade Runner*, Rick Deckard is a hunter of synthetic humans, called *replicants*, who falls in love with one of his prey before discovering that he himself is a replicant. The contradiction encourages us to rethink what it means to be human in the first place. That we comfortably apply the label *ironic* to each of these cases suggests that there is a shared essence that unites them all, something akin to a familiar surprise that is apt and inappropriate at the same time.

GOING VERBAL

When it is used with skill, verbal irony can deliver an incisive critique with all the mocking intent of sarcasm and all the plausible deniability of a happy accident. It thrives on ambiguity and resists easy labels, so it is scarcely surprising that when the hashtag *#irony* is used to harvest a data set from Twitter, our haul includes almost as many kinds and folk theories of irony as there are uses of the tag. While some uses naturally cluster into the situational, the verbal, and the dramatic, many more suggest a close relationship with hypocrisy, coincidence, and poetic justice. This odd kinship among competing viewpoints is also evident in formal theories of irony, since the big tent of irony research rests on three poles that complement as much as challenge each other: opposition, echo, and pretense.

The first tent pole comes closest to capturing our folk intuitions of what verbal irony is all about—saying *A* while meaning or implying *not-A*—so it is this sarcasm-adjacent view that has held sway for the longest time in the field of pragmatics. This is the field that concerns itself with why we say one thing, such as, “Can you reach the salt?” or, “Aren’t you

cold?” when we really mean something else, such as, “Pass the salt,” or “Close the bloody window.” In the implicature account of H. P. Grice, the influential philosopher of language for whom the Gricean school of pragmatics is named, irony marries opposition with context so as to indirectly imply what we really want to say.⁷ But why not just say that directly?

Imagine that you and a friend are walking in a dubious part of town when you see a car jacked up on bricks, its wheels missing, its windows shattered, and its stereo plundered. If you turn to your friend and declare, “Well, this is a *great* place to park,” the context makes it obvious that you really mean the opposite: “What a *stupid* place to park.” Grice’s work explored the cooperative principle that tacitly guides our dealings with others whenever we use words. His *maxim of quality*, for instance, captures our unspoken goal of contributing truthfully to an exchange by refraining from saying that which we believe to be false.⁸ His insights concern the general tendency of speakers toward relevance and plainspoken truthfulness. Although they are formulated in the guise of injunctions and rules, his maxims are not hard rules but soft norms and are just as informative when they are flouted.

To flout a maxim is to violate it in a recognizable way. When I say that “this is a great place to park” in a spot where I know that crime is rampant, I am flouting Grice’s maxim of quality, but not without reason. If my aim is to deceive, then I will want my violation of the maxim to go unnoticed. But if my aim is to be ironic, I will want my flouting to be recognized, at least by that part of the audience I am trying to reach. Ironists benignly violate the maxim to reveal the truth of a situation that another person, perhaps wholly imaginary, fails to grasp. In our car example, we mock the naiveté of a driver who believes this to be a safe place to park. Flouting is a benign kind of violation that sets out to inform rather than to deceive.

We can easily picture that hapless driver swatting away any concerns about his choice of parking space. “This is a great place to park,” we can imagine him saying, making our ironic observation a mocking echo of his lack of foresight. As Deirdre Wilson and Dan Sperber and their many supporters have argued, irony echoes that which it seeks to mock, thereby transplanting a failed prediction from a context in which it seems sensible to one in which it seems anything but.⁹ Irony even allows us to echo an

entirely imaginary claim that was never made at all, provided we can still reasonably attribute it to the target's personality and beliefs.¹⁰ Since the driver's choice speaks for itself, it is fair game for our echo. Yet when echoes are as loosely characterized as this, you might suppose that any claim at all can be interpreted as the echo of an imagined remark in another context. As an ironic echo veers from the actual to the distorting to the imagined, it demands a greater suspension of disbelief and a greater openness to pretense from an audience.

So when does an imagined echo tip over into full-blown pretense? We may not know anything at all about the owner of that vandalized car or his reasoning, but we can pretend that we do and put into words the kind of naive thought process that would lead to the same outcome. In the pretense view of irony championed by Herbert H. Clark and Richard J. Gerrig, ironic speakers don't so much echo someone else as momentarily pretend to be someone else.¹¹ Just as an echo can be entirely imaginary, so can the injudicious person that the ironist pretends to be.

What matters is that the pretense is recognized for what it is, an attempt to get inside the mind of another so as to criticize what is found there. The pretense can divide an audience into two parts: those who see through it versus those taken in by it, so that the ironic utterance becomes a critique of one by the other. Mozart's replies to Salieri can be understood as this kind of face-saving pretense, one that allows each man to speak his own truth while pretending that no offense has been given or taken. However, an effective pretense requires us to play our parts well, which requires the pretender to say, mention,¹² or echo¹³ what is expected of the role. So it is hardly surprising that what looks like play-acting to an adherent of the pretense account often looks just like an echo to supporters of the echoic theory.¹⁴

In fact, while some uses of irony are obvious examples of the echoic theory at work, as when we mockingly repeat another's failed predictions verbatim, others are clear instances of satirical pretense, as when we hijack the distinctive voice of another to say exaggerated things that they would never say. Since a one-size-fits-all account forces us into mental somersaults to squeeze the problem cases into a single container concept, it just seems easier to view irony as a polythetic concept that doesn't obey any single

pattern, but is instead captured by a family of related patterns. If we can agree that irony frequently sounds like a mocking echo, sometimes resembles a playful pretense, and occasionally reduces to contrariness and opposition, then why not train a machine to apply all three viewpoints in parallel, with each module applying its own pragmatic assumptions? In cases that are paradigmatic of a single view, just one module might express confidence in its judgment, while uncertain edge cases might cause all three to blare at once, like the alarm of a car whose windows are all blown out at the same time.

FIRE AND RAIN

Our ability to communicate with sarcasm and irony goes hand in hand with our ability to recognize when others use them too, not least because we need to put ourselves in the audience’s position whenever we aim to be playfully insincere. These detection and generation abilities develop in tandem: as we engage more with nonliteral language, we gradually learn to do unto others what has already been done unto us. But for machines that lack a social life, the connection between detection and generation is much less developed. A detector of irony and sarcasm in online reviews may have no generation capabilities at all, while a Twitterbot that generates ironic tweets may have no appreciation of the playfulness of the responses it garners from human users. Programs with limited goals have limited needs, and so we can train a statistical detector of sarcasm and irony on a large corpus of annotated examples with no concern at all for how our detector might later be used to generate. Those explicit annotations—*this* is sarcasm, *that* is not—make this a form of “supervised learning.”¹⁵ Fortunately, our propensity to mark out sarcasm and irony on social media, to avoid public shaming should it all go terribly wrong, makes it a snap to gather plenty of supervised learning material.

As the textual equivalent of a mocking tone of voice, the #sarcasm hashtag on Twitter (just like the /s tag on Reddit) allows social media users to telegraph their mocking insincerity, as in the following lightly edited and anonymized tweets:

I love it when an atheist tells me what the church I belong to believes!! #sarcasm

I hate it when cars go too fast in racing! #sarcasm

I love it when clients try to tell me how to do my job. #sarcasm

I'm thrilled that forward-thinking and inclusive policies have made changing rooms so safe for my daughters. #sarcasm

I hate it when actual videos play between my YouTube ads. #sarcasm

These examples are as paradigmatic as sarcasm gets on Twitter. A disingenuous attitude frames a situation that will likely evoke a very different response from an informed reader, making “#sarcasm” a convenient shorthand for the disclaimer: “I don’t mean this. How could *anyone* mean this? I mean the exact opposite!” The negation is more explicit in the teenage fondness for an utterance-final NOT, as evident in this dual response to @realDonaldTrump and @KanyeWest on Twitter: “Two stable geniuses. #NOT.” A sentence-final negation revels in the opposition at the heart of irony and sarcasm, but it is not in itself ironic or sarcastic. In contrast, other markers, such as the tag #yeahright, are self-contained nuggets of formulaic sarcasm. If #sarcasm is theory neutral, and #NOT embodies Gricean opposition, then #yeahright has a clear affinity to the echoic mention theory of Sperber and Wilson. As demonstrated by “Travel with the kids, they said. It’ll be fun, they said. #yeahright,” and “You’ll be safer in Los Angeles. #yeahright,” the hashtag appears to agree with and simultaneously reject a view previously expressed by others.

So many sarcastic utterances have an obvious fire-and-rain structure that our machines can learn to discern for themselves if given enough cases to analyze. The fire of a strongly worded attitude (e.g., “I love,” “I hate”) is doused in the cold rain of a disproving proposition, to which an explicit tag like #NOT or #sarcasm is an additional, perhaps redundant, wet blanket. We just need to teach a machine to first detect fire, and then rain, in the same utterance. To this end, Ellen Riloff and her collaborators designed an iterative process to allow a machine to bootstrap a robust appreciation of sarcasm by first learning from the simplest cases.¹⁶ When a tweet tagged with #sarcasm begins with “I love . . . ,” we can be confident that what

follows is a most unlovable situation. By harvesting any tweets of the form “I love <situation> #sarcasm,” a machine can learn to characterize the contexts that are most deserving of sarcasm. Once it acquires a statistical model of these situations, the machine can shift its focus to the range of attitudes, other than *love*, that users of sarcasm tend to evoke. So, by assuming a little about fire, it can learn about rain, and then use its understanding of rain to better understand fire, and so on, until it has a firm grasp of fire and rain and of how they come together in the same texts.

Different sources of online text provide this mixture of fire and rain in their own ways. Online product reviews, for instance, often come with star ratings that summarize the content of the user’s text with a numeric score. If the average sentiment of the text is out of kilter with its assigned score—as when a highly positive(-seeming) text is saddled with a one-star rating—then either a mistake has been made or the writer is being insincere. As with the #sarcasm tag, a star rating provides the ground truth against which a text can be understood, allowing a machine to untangle its propositional attitudes from its propositional content. This makes online product reviews a rather obvious focus for research in sarcasm detection, not least because of the commercial benefits that accrue to e-retailers.

So it was from online reviews that Oren Tsur and his colleagues at Jerusalem’s Hebrew University took the training data for their sarcasm detector, SASI (semi-automated sarcasm detector).¹⁷ Using sentences that had been annotated for sarcasm, they sought to find recurring patterns that would also cover future uses. High-frequency words such as determiners, prepositions, and punctuation marks were replaced with a generic [HFW] marker, while low-frequency content words such as adverbs, nouns, and verbs were replaced with [CW]. Proper names were also replaced with a marker such as [company] or [product]. When a pattern that matches a clear case of sarcasm also matches a clear case of nonsarcasm, it is rejected as undiscriminating. The patterns that remain, such as “[company] [CW] does not [CW] much,” can now be viewed as the syntactic fingerprints of sarcasm in an online review: not enough to convict on their own, perhaps, but enough to confidently bring a charge when they are present in the aggregate.

For each sentence that SASI analyzes for sarcasm, its stock of syntactic patterns defines a feature space for a vector representation. If there are n patterns, then each sentence is turned into an n -dimensional vector, where the value of its i th dimension corresponds to how well the sentence matches the i th pattern. For each pattern, SASI checks whether it matches a sentence exactly, loosely (allowing for intervening words), partially (allowing for missing words), or not at all. It assigns a value of 1.0 to the corresponding vector dimension for exact matches, 0 for nonmatches, and a real value between 0 and 1 for loose or partial matches, depending on how loose or partial they are. It then compares the newly built vector to the ones it has already built for the positive (sarcastic) and negative (nonsarcastic) examples in its training data. By measuring the Euclidean distance between the new and old vectors, SASI assigns a similarity score to every possible match. The greater the distance, the lower the similarity, and vice versa.

In this way, SASI can find the exemplar in its training set that best matches the sentence under investigation, at least as far as its understanding of sarcasm goes. But SASI needs to do more than find a single best match. By finding the k nearest matches to the current sentence—using an algorithm called k -NN, for k -nearest neighbors—it obtains a nuanced comparison set. Now it can, for instance, select the most common label (sarcastic or nonsarcastic) in this set of k comparisons. What SASI does is a little more sophisticated than this: it calculates the majority label in its set of k matches, but also takes account of the frequency of those labels in the training set as a whole. Since sarcasm is far less frequent than nonsarcasm, matches labeled *sarcastic* should be viewed as weightier pieces of evidence. So a new sentence is deemed sarcastic only when the weighted average of the numeric labels assigned to similar sentences from past reviews tips the scales to sarcasm.

AI systems are defined by the choices they make. Of the many ways in which a machine can slice, dice, and weigh the linguistic features of a text, each commits to just a few. While systems like SASI learn from real examples, they are limited in what they can learn, and how they learn it, by some restrictive assumptions about what marks out a sarcastic utterance from a nonsarcastic one. So why not let the learning algorithm decide for

itself which features to use and which weights to assign to each? This is the guiding philosophy of connectionist approaches to sarcasm detection, and of many other applications of text classification too. If it is given a sufficiently rich representation of words, a neural network can learn to discern, in a supervised fashion on a large training set of annotated exemplars, which combinations of features to reward and which to ignore. So, in the encoding stage of a neural model, the sequence of words in a text is converted into a vector of numeric values, allowing the network to discover discriminating features and patterns that engineers might never find, much less understand, for themselves.

As we saw in chapter 6, an artificial neural network is a densely connected graph of simple computing units, each of which applies a mathematical function to the output of earlier units, to provide a value that can then feed into units further up the network. In this way, an initial layer of input values is mapped, via a succession of scaling, aggregating, and thresholding operations in the units of intervening layers, to a final layer of outputs. Deeper neural networks have more layers and provide more opportunities to transform an initial vector encoding of the input into a series of incrementally more abstract, concise, and insightful features.¹⁸

When the input to a neural network is an image—suppose that our network must tell cats from dogs or happy from sad faces—the initial encoding is obvious: we can simply take a 2D array of pixel values and flatten it into a 1D vector with the same total number of values, so that a 100×100 pixel image becomes an input layer of 10,000 numeric values. A bank of filters called a *convolutional layer* then transforms those values into sets of local features that represent edges, corners, or changes of brightness, and propagates the newly extracted features further up the network.¹⁹ But when the input to our network is not naturally represented as a list of numbers, as when we are dealing with the words of a text, we need a means of encoding each word as a vector of continuous values. In the previous chapter, we met several vector-space approaches to word meaning, such as Word2Vec, GloVe, and LSA, that do just that. When we use these vectors as inputs, convolutional layers do surprisingly well at extracting regularities from their seemingly unfathomable representations. In fact, a neural

network can generalize over words it has never seen before if it has encountered words with similar meanings and encodings.

One of the first neural network models for sarcasm detection was presented by Aniruddha Ghosh, who called his approach *fracking*.²⁰ Aniruddha, who was my PhD student at the time, wanted to capture the idea that his network was cracking open an utterance to release the bitter sentiment bubbling beneath its surface. His network aims to detect sarcasm in tweets and accepts as its input a sequential concatenation of the vector encodings of each word in a given text. The numeric features of this encoding, in which no single value is meaningful in itself but makes sense only in the aggregate, are piped through a series of convolutional layers that extract low-level features from the tweet. A dropout layer encourages the model to generalize well to unseen data by randomly discarding a set percentage of features along the way. The features that survive then pass through a pair of LSTM (long short-term memory) layers that attune to contextual dependencies between nonadjacent words in the input, such as the *fire* words near the beginning of a tweet and the *rain* words near its end.²¹ The final layers of the network eventually boil the feature space down to a single, all-important decision: Is this input sarcastic or not?

Modern AI tool kits make it easy to build and train neural models such as this.²² Aniruddha demonstrated that his particular assemblage of off-the-shelf network layer types—which retains a certain elegance despite being the AI equivalent of a Rube Goldberg device—performs rather well on actual data, whether on the data sets we gathered and annotated ourselves or on public sets annotated by others, such as those of Riloff and Tsur. It has a recall of .899, which means that, on average, it finds nine of every ten sarcastic tweets in the test data. It also has a precision of .904, so nine of every ten tweets that it deems sarcastic will have been labeled as such by human annotators. Since Aniruddha published his results, this performance has been exceeded by other models from other researchers, but the general principle remains the same. A network must map its textual inputs to a vector encoding from which it learns to extract and aggregate the features that allow it to partition the universe of sentences into sarcastic versus nonsarcastic.

While Aniruddha is a capable speaker of English, he is not a native speaker, and he diverged markedly from other annotators on certain kinds of insincere tweet. For instance, he detected no implicit negation, and not a hint of sarcasm, in tweets marked #yeahright, while others easily detected the cynical shrug behind the tag. In truth, even native speakers have trouble detecting sarcasm after the fact,²³ in tweets that lack the context of a motivating time line.²⁴ So why not ask the authors themselves to annotate their own tweets, to give us privileged access to their true intent? Aniruddha first built a baseline detector to find and analyze responses to tweets from specific magnets for sarcasm, such as divisive celebrities, comedians, and political figures.²⁵ When the detector deemed a tweet to have a reasonably high likelihood of sarcasm, an automated Twitterbot would then reply to its author to ask whether that person had set out to be sarcastic. The bot's addressees didn't appear to view this as an invasive use of the platform, and most seemed happy to reply with a confirming "yes" or disconfirming "no" to its curious inquiries. These responses then allowed Aniruddha's neural network to learn about sarcasm from a training set annotated with the ground truth of its authors' actual intentions.

Sarcasm and irony are the ultimate turd blossoms, acts of playful criticism that grow in a motivating context of disappointment and negativity. We can do better at detecting sarcasm when we are privy to the context in which it blooms, and on Twitter this is approximated by a user's time line. There we can find valuable clues to the personality and current mood of a Twitter user in the user's most recent tweets: Does the user appear angry and remote, or upbeat and jovial? When one tweet is a reply to another, the first might inform us as to why the second is sarcastic. So why not give our sarcasm detectors access to these insights too? In a subsequent revision to his network, Aniruddha expanded his input layer to accommodate two new kinds of context. When the network analyzes a tweet that is a reply to another tweet, the vector encoding of that prior text is presented to the network too, so that its memory layers can learn to recognize dependencies across the tweets. For the second kind of context, Aniruddha would also need a way to vectorize people.

Sarcasm detection is a specialized form of sentiment analysis.²⁶ If it is applied to a large enough body of text, this analysis can unearth fascinating insights about an author too. For instance, the use of personal pronouns such as *we*, *us*, and *our* relative to the use of *I*, *me*, and *mine* can tell us the extent to which a user is group- or self-focused. Certain negative-affect words indicate hostility, while others suggest worry and depression. Positive-affect words can indicate an upbeat mood, while the use of other people's names and Twitter handles can tell us whether a user is socially aware and well connected. Logical connectives and long sentences (with perhaps long or uncommon words) can indicate intelligence and analytical tendencies, while time-related words like *now*, *soon*, and *today* can tell us that a user is writing in the moment. If sentiment analysis can reduce a user to some core psychological dimensions—after all, this is what human *shinks* do—it might be worthwhile to also feed these dimensions into our neural networks.

AnalyzeWords.com is a popular sentiment analysis and author-profiling tool that uses the LIWC (linguistic inquiry and word count) resource to provide an eleven-dimensional analysis of a Twitter user, based on the text of that person's most recent hundred or so tweets.²⁷ So in addition to the immediate prior context of a given tweet, Aniruddha piped the numeric values of these eleven dimensions into his neural model, as calculated by AnalyzeWords for the preceding thousand or so words from the author's time line. In effect, he was allowing his model to make judgments not just about the mood of a given tweet, but about the mood of the person who wrote it at the time it was written. The architecture of the resulting multi-input network is illustrated in figure 9.1.

By giving the sarcasm detector the prior setup tweet to a potentially sarcastic utterance, the network is better able to discern the intent of its author, since it can now find something akin to an incitement for a caustic response. By also giving it a coarse sense of who the author might be, psychologically speaking, the network can motivate any sarcasm as an extension of the author's current mood. Depending on the precise configuration of the neural network—these architectures encourage a great deal of fiddling, so variants abound—this prior context tweet is worth an extra

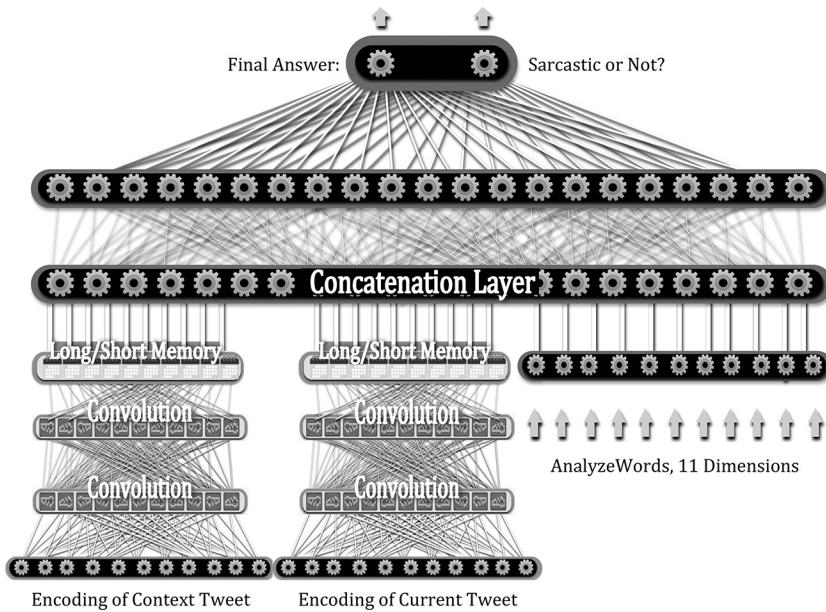


Figure 9.1

A neural architecture for sarcasm detection that includes two kinds of context data.

6 to 7 percent of detection accuracy to the system. Additionally, the recent mood of the author, up to and including the current tweet, adds 6 or 7 percent in combined precision and recall, again depending on the network configuration that is used. Sadly, the benefits of each kind of contextual information are not additive. When a network is given the prior context tweet *and* a profile of the author, we see much the same bump in performance as when we give it one or the other. The network can get what it needs from either of them, but it doesn't seem to need them both.

THE MARK OF @CAIN

Explicitly tagging an utterance on social media with #sarcasm or #irony can seem as graceless as stapling a receipt to the front of a gift-wrapped birthday present. Yet as much as we might decry the slow death of nuance in modern communication, social media users have their reasons. Indeed,

even in normal discourse, we have always had ways of marking our utterances to hint at the insincerity within. We can find these markers in all registers of speech and at all levels of literature.

Let's start with the least subtle, which can also be the most effective. Do you ever make air quotes while speaking? Since this faintly ridiculous rabbit-ears gesture marks the part of an utterance that should be understood as an echoic mention of someone else's words and ideas, it also serves as a marker of irony. Just think of it as the pantomimic realization of Sperber and Wilson's echoic theory of irony that tells an audience exactly where to look when considering a speaker's sincerity. Air quotes nicely complement an arch tone of voice, but in settings that are neither face-to-face nor hands-free, we may have to rely entirely on the latter. Consider this extract from the 2015 film *Anomalisa*, as written by Charlie Kaufman.²⁸ Here we join a conversation between Michael, a jet-lagged traveler, and the chatty taxi driver who is taking him to his downtown hotel:

Michael: Is there a toy store near the hotel?

Driver: A toy store? You mean a TOY store?

Michael: Yeah toy store.

Driver: Yeah there's a TOY store about two blocks up. It's open all night.
Real class place.

We don't expect a driver to take his hands off the wheel to mime air quotes for the benefit of a passenger. As listeners, we infer that spoken emphasis calls for special analysis, to reach what Rachel Giora calls a *non-default* interpretation of a familiar form.²⁹ The driver, believing Michael to have a seedy, nondefault interpretation of "toy" in mind, directs him to a retailer of sex toys while mocking his apparent coyness with echoic irony. But Michael really does have a default interpretation of "toy" in mind. Befuddled by jet lag, he buys a Japanese sex robot for his young son.

The written form of the air quote is the *scare* quote. While it is just as obvious, it is also just as ambiguous. Authors use written quotation marks not just to quote others, but to highlight words and phrases that are deserving of special attention, especially when they are used in nondefault ways.

Imagine that you come across a blackboard in an ice cream parlor that advertises itself as a purveyor of “gelato.” The quotes do more than mark out an Italian word in a non-Italian setting. Rather, they seem to appeal to an invisible Italian authority to proclaim “We don’t just sell ice cream, which Italians call ‘*gelato*.’ We really do sell *Italian ice cream*, of the kind Italians would be happy to call ‘*gelato*.’” On a hot day, the quotes encourage us to reach for our wallets all the faster. Conversely, we might run a mile when the *chicken* of *chicken nuggets* is wrapped in scare quotes on a fast-food menu. While this use also appeals to an invisible authority, who can take encouragement from the nondefault interpretations of health inspectors and corporate lawyers?

The more affectively charged the word in scare quotes, the more caution they demand. When Gore Vidal tartly described Andy Warhol as “the only ‘genius’ I’ve ever known with an IQ of 60,” his use of “genius” positively dripped with scorn, echoing as it did the sensibility of an art world in thrall to modernity. Ironic scare quotes draw our attention to an echoic mention while scaring us away from its default interpretation. In this case, they shoo us over to Vidal’s perspective, where the Warhol mystique is better explained by lower rather than higher intelligence. In truth, they are scarcely needed in utterances like Vidal’s, given his penchant for acid barbs, yet they are so easy to use that it would be remiss of a machine not to avail of them when communicating its own ironic observations.

A metaphorical Twitterbot, @MetaphorMagnet, uses techniques like these to package its own figurative views on everyday concerns. The bot is fueled by a trove of propositional content that it mines from other figures of speech on the Web, such as similes.³⁰ By observing the world as it is refracted in its Web data, @MetaphorMagnet devises its own viewpoints and frames them in a wide variety of different forms. Consider the bot’s ironic lament on a topic close to its heart:

#Irony: When some playwrights use “inspired” metaphors the way programmers use uninspired hacks.

The bot follows a belt, suspenders, and safety-pin approach to framing its ironic tweets. First, since each of its metaphors is situated in the overlap

of two scripts, such as its understanding of *plays* and *programs*, this juxtaposition is made explicit via the construction “When [X] the way [Y].” Second, the disputed quality that the irony both foregrounds and undermines is wrapped in scare quotes. Third, abandoning all subtlety, the #irony hashtag is finally stapled to the front. @MetaphorMagnet generates dyspeptic observations like this one all day long, despairing that “some” instances of a familiar idea fail to live up to the expectations it acquires from the Web. It imbues its tweets with an air of regret by picking only positive qualities to highlight in quotes, so that readers are scared away to a negative interpretation.

In fact, this is how we operationalize the success of an ironic communication in @MetaphorMagnet: an ironic metaphor works as irony if it causes an appreciable downward shift in the reader’s perception of the sentiment of the disputed quality (that is, the word in scare quotes). If we ask unbiased judges to rate the sentiment of a word such as “inspired” in a null context—that is, the word on its own with no surrounding sentence to bend its meaning—we can expect them, on average, to rate it very positively on a scale from -1.0 to +1.0. But if we now ask other judges to rate the same word in an ironizing context, such as the tweet above, we should expect their averaged sentiments to show a significant downshift to the negative.

By operationalizing irony in terms of its effect on perceived sentiment, we can experimentally measure the relative success of our different marking strategies. So, how much do we gain from stapling a painfully obvious #irony tag to the front of a tweet? Are scare quotes really as scary as they’re cracked up to be? And what is the value of highlighting both of the overlapping scripts instead of just one? My colleague Alessandro Valitutti and I set about finding answers to these questions on the crowdsourcing platform Crowdflower (since renamed Figure Eight), by paying volunteers a small sum to rate the sentiment of different qualities in one or more structural variants of @MetaphorMagnet tweets.³¹ We asked the bot to whip up eighty of its ironic tweets and to package each in a variety of these variant forms.

As exemplified by the “inspired” tweet, the all-in form combines script opposition with scare quotes and a hashtag. A series of variants is also generated by ablating one or more of these markers. Some judges see only the base assertion, which, for the tweet above, is “Some playwrights use

inspired metaphors.” An equal number see only a script opposition variant, such as, “Some playwrights use inspired metaphors the way programmers use uninspired hacks.” Others see one of these variants with either a front-loaded #irony hashtag or scare quotes, and yet others see a variant that has both of these additions. This set of variant forms, for which we recruited an equal number of raters, allows us to tease apart the relative merits of each marking strategy by measuring the average downshift in sentiment for each.

It turns out that the #irony tag makes no significant contribution, statistically at least, to the appreciation of ironic intent. It may be that we just ignore it as empty posturing, or perhaps our folk theories of irony are so diffuse—forced, as they are, to explain verbal, situational, and dramatic irony—that the tag has been drained of any value as a marker of sentiment. In contrast, explicit script opposition has an appreciable effect, producing an average shift in sentiment of .22 to the negative. Scare quotes also have a clear effect. If used as the only ironic marker, they yield a mean downshift of .10, but their effect is additive. So when script opposition and scare quotes work together in the same tweet, we see a mean shift of .31 to the negative. These markers are the devil’s seasoning: although they are not negative in themselves, they really do bring out the sour tang of a rueful observation.

Markers of ironic intent exhibit a little more subtlety in the world of literature. Consider this image from the modern master of the simile, Raymond Chandler:

Even on Central Avenue, not the quietest dressed street in the world, he looked about as inconspicuous as a tarantula on a slice of angel food.

*Farewell, My Lovely*³²

Chandler uses “about” to mark his comparison of Moose Molloy, a white man (“he”), to a black tarantula. He revels in Molloy’s conspicuousness, and seems to mock him for his failure to be otherwise. Not only does the “about” imply negation here, since *about* [X], *almost* [X], and *nearly* [X] all entail *not* [X], it also implies failure. When negation collides with mockery and pretense in this way, can irony be far away?

The modifiers *about*, *almost*, and *nearly* are the subtle speaker’s answer to Twitter’s in-your-face #irony and sullen #NOT tags. We first met them

in chapter 3 and can find them used in creative similes all over the Web.³³ As understated markers of irony, they work much like road signs for hairpin bends, alerting us to wild turns ahead. They remind us that conventional speakers use only one side of the road when looking for meaning, but creative speakers and listeners use both.

So, how useful are markers like “about” at signaling irony? While few linguistic markers have the determinacy of a hashtag, these markers still have a clear effect. Even when a marker is not overtly present, the ghost of one might be. Think about the witty similes that you use from time to time. Do you mint each one yourself, or do you often repeat something that you picked up elsewhere? A study of creative similes on the Web finds a long tail of one-off creativity and a fat rump of reusable gems that have acquired an almost proverbial status. The folklorist Archer Taylor labeled the most common of these “proverbial similes.”³⁴ Even if you don’t use a simile in its “about” form, others before you may have, so when you search on the Web for other uses of the same simile you may find that the “about” form is the dominant one.

When my PhD student Yanfen Hao and I set out to automate the classification of creative similes as ironic/sarcastic or not, we based our classifier on three basic questions. The first asks whether the quality ascribed by a simile to its target is more desirable than undesirable. Is it more positive than negative? The second asks whether the simile’s “about” form is the dominant form on the Web. This can be answered by using Google to seek out and count its various uses. The third asks whether there is any evidence that people can say much the same thing with linguistic constructions that are less amenable to irony or sarcasm than the simile form. If so, a speaker’s intent is very unlikely to be ironic in the case of the simile form.³⁵

For instance, the “[X][Ys] such as [Zs]” pattern allows us to ascribe a quality [X] and a category [Y] to an idea [Z] with a linguistic construction that is almost never used ironically. When we read “hot places such as saunas and kitchens,” we can be confident that the writer really does believe that kitchens, like saunas, are hot. Patterns like this—named “Hearst patterns” after Marti Hearst, one of the first to use them as retrieval filters—are a reliable basis for mining commonsense knowledge from corpora.³⁶

The frame “as [X] as [Y]” is a productive Hearst pattern that allows us to mine stereotypical associations from Web similes on an industrial scale. So if we want a machine to know that jelly is wobbly, dogs are loyal, and snails are slow, this is an expedient way for it to acquire such beliefs.

This is exactly what Yanfen and I set about doing. By firing a barrage of Hearst-like queries at the Google search engine, we hoped to amass a diverse knowledge base of stereotypical views on everyday things. We used a pair of complementary Hearst patterns that work much like the pairing of a straight man and a funny man in a comedy duo. The simile pattern “as [X] as [Y]” retrieves a large body of stereotypical beliefs, but it also pulls in a surprising amount of humorous content. It turns out that 15 to 20 percent of the similes we find are ironic, and many more are playfully exaggerated. Yet because the “[X][Ys] such as [Zs]” pattern has a straight man’s reduced tolerance for irony and humor, we can use it to check whether a simile’s meaning is also found on the Web in this more serious form. When the same viewpoint can be harvested with both patterns, we know we can take it seriously.

Since our goal was to amass a large knowledge base of reliable commonsense perspectives, we initially treated any unserious similes as noise, or a kind of waste product to be thrown away. We certainly didn’t want our machines to believe that bowling balls are hairy, rabid dogs are friendly, and Irishmen are tanned. Yet if your goal is the analysis of humor, absurdity is not so much the bathwater as the baby. We want our computers to move freely in this comical upside-down world where sense looks like nonsense and every insight wears oversized clown shoes.

The pairing of straight man and funny man works so well because humor bends straight norms into absurd forms. If a machine is to be straight *and* funny, we must allow it to see the world as fact and fabrication. Since Web similes vividly show how we humans do both, we should feed our machines as many as we can, helpfully labeled as straight or ironic. In addition to those built from stereotypical norms, we also find many similes that subvert those norms, as well as thousands of other “about” forms that nonironically use these norms in comical ways. What we find in the noise is a golden treasury of perverse mental imagery in which, as we saw in chapter 3, even the fool’s gold has reuse value for automated humor.

WHAT? ME WORRY?

Winston Churchill defined diplomacy as the art of telling others to go to hell in a way that makes them ask for directions. When used with aplomb, irony requires an emotional intelligence in how we pick our words so as to be true to ourselves and the feelings of others. This turns irony into one of the most useful tools in the diplomat's rhetorical toolbox. In the end analysis, it is a tool that we must also put within the reach of machines if they are to reconcile the facts of a situation with the human feelings of their users. By and large, we still view machines as a source of frustration when things go wrong, not as partners with whom we can share our disappointments. While programs that help us to write new jokes remain of niche value, the inexorable march to replace human services with Web-based chatbots offers a compelling use case for automated humor in human-machine dialogue.

We met two of AI's most famous chatbots, Eliza and Parry, back in chapter 4. Each bot offers ample scope for creative interaction with its human users. Eliza could surely find use for ironic understatement, while cutting sarcasm is a natural fit for Parry's unstable and accusatory personality. But neither bot pursues these avenues for its own peculiar reasons. Eliza lacks the diverse set of expectations on which irony crucially relies, and so it has no real sense of anything beyond its own shallow templates. Parry, in contrast, does have some knowledge of the narrow slice of the world on which it fixates, but this is neither broad enough nor deep enough to support a general capacity for sarcasm. Since the bots never learn, they cannot acquire from their users that which was denied to them by their designers. Although modern chatbots make use of machine learning, they still contain more than a little of Parry and rather a lot of Eliza. Like Parry, chatbots designed to talk to specific concerns must possess enough domain knowledge to drive a dialogue forward. And when a user drifts outside a bot's limited area of expertise, generic Eliza-like trickery is still needed to nudge them back onto safer ground.

So what would a chatbot with a sense of irony and sarcasm look like? Crucially, it must look as if it has an imagination when it answers the big questions head-on. An ironic bot might still deflect some topics it cannot handle, but will typically do so with grace and occasional wit. For the topics it is able to tackle, an ironic bot won't hide behind vague generalities, but will

actively move the discussion from the generic to the specific, evoking vivid images with provocative turns of phrase. Like Parry, it will eagerly inject its own ideas into a conversation, but like Eliza, it may also use the shape of the input—rather than its actual meaning—to formulate a reply when words or ideas fail it. This allows the bot to move with ease between serious questions (“What is life?”) and silly banter (“Why is life like a banana?”).³⁷

My own group’s efforts to build a bot like this focus on Figaro, a conversational system that blends Eliza’s shallow tricks with Parry’s knowledge-driven fixations. Figaro does not fixate on a particular set of topics in the way that Parry’s world is defined by its fascination with horse races, gambling, and gangsters, but strives for a figurative treatment of any topic in its database of norms and stereotypes, which it acquires in its large-scale trawls of the Web. Figaro also uses the Google n-grams to build a metaphorical picture of topics raised by users. So, on the topic of life, 4-grams like these supplement what the bot already knows:

life is a *highway*, life is a *gift*, life is a *journey*, life is a *game*, life is a *mess* . . .

These give Figaro a handle on topics that can be difficult for humans to pin down. By moving quickly to specifics—for example, from breath to *breath of fresh air*, or from rule to *rule of law*—Figaro injects concrete ideas to drive a conversation forward:

User> What’s life all about, Figaro?

Figaro> Some compare life to the rule of law.

User> why?

Figaro> Maybe worrying about life will have you fighting for justice?

User> how so?

Figaro> Worrying about life is as threatening as an ominous storm.

Figaro, like Parry, has its own agenda and seeks to maneuver its users into a space in which it can contribute to, or even lead, a conversation. This space is not topic specific, but is, rather, the space of storytelling in general. The bot’s built-in story-generation system (from chapter 4) gives it a basic

understanding of cause and effect, which allows it to predict or suggest what happens next in a user's tale. Figaro takes every opportunity to nudge the conversation onto the causal tracks of its narrative system, seemingly pulling topical story ideas from out of left field:

User> why is life like the Turing Test?

Figaro> You feel as challenged by life as by the Turing test.

User> go on.

Figaro> Suppose that you were retiring replicants with Rick Deckard?

Of particular interest to Figaro are the emotional resonances of words, ideas, and the metaphors it finds in the Google n-grams and in the wacky similes of the Web:

User> why is my life so much fun?

Figaro> Because your life is a joke.

User> why is that?

Figaro> Your life is more amusing than a bag of wet shit.

User> not true

Figaro> You feel as shocked by your life as by a child's dirty joke.

User> I do not

Figaro> Many lives are as shocking as a child's dirty joke and just as sincere as a fart joke.

If these fragments of dialogue suggest a human user being pummeled by Figaro's viewpoints, then so much the better. Users can always assert themselves, and it is important to show that Figaro can do likewise, with insight and imagination. As to whether a system should ever use words like *shit* and *fart* with the abandon it exhibits here is a matter of some concern, as it has learned these formulations from Web content that is home to much worse. We tackle the issue of self-censorship in the next chapter, to explore where—and, crucially, *how*—the line should be drawn.

10 AT WIT'S END: LESSONS FOR THE FUTURE

WALK THE LINE

In 1972, George Carlin famously exploited the distinction between what a comic can say on the stage of a club and what is considered acceptable on TV. His classic routine took delight in listing “the seven words you can never say on television,” and Carlin invested the words—“shit, piss, fuck, cunt, cocksucker, motherfucker, and tits”—with all the rhythm of a holy mantra.¹ Although his list was not nearly as authoritative as he implied and had no basis in regulatory standards, it sounded right. These really were words you could never expect to hear on TV in the 1970s. And while more could have made his list in 1972, such as *bitch* or *bastard*, any additions would only have derailed his analogy with the Bible’s seven deadly sins.

Carlin’s star turn is remembered as a high-water mark in the history of comedy, but his violation of taboos would surely have offended as many as it delighted. Not everyone shares the same view of what is benign and what is not. In this case, the hazardous materials come clearly marked with danger signs, but comedians cause offense with more innocent words and ideas too. We can chalk this up to personal taste when talking of human comedians and their efforts to reach an audience, but automated humor is different. It has the potential to be pervasive *and* invasive, to intrude into our lives in surprising ways whether we want it to or not. Automated humor generators can run amok on social media, where they may be weaponized by bad actors to target third parties or entire social groups, but the downsides are more insidious still: as humorous AI becomes a selling point

for our favorite apps, it gains the power to exploit our personal data in the most egregious of ways.

This chapter looks at the potential dark side of automated humor to explore what can happen when the Sith hits the fan. Only by imagining what can go wrong can we build real safeguards into our systems to mitigate the worst excesses of AI. The surest way to prevent abuses is to imbue our AIs with a moral imagination of their own so they can predict for themselves the harm they might cause to others. However, we can hardly wait for such a grand ambition to come to fruition, and in the meantime, there are simpler and more practical steps that we can take. If an ounce of prevention is worth a pound of cure and a ton of abject apologizing, then this closing chapter offers some clear lessons for some predictable problems. In that spirit, we begin where we intend to end up: in someone else's head.

No two people have exactly the same intuitions, or the same implicit models of notions like appropriateness. When we set out to be funny, we must assume that our model is also a pretty decent approximation of the models of others. This is what it means, more or less, to say that different people share a sense of humor. The assumption allows comedians to foresee how audiences will traverse a joke's surface, to predict where they will stumble and how they will feel when they do. A joke that falls flat, or that hits the wrong target, may be impossible to walk back, so comedians are advised to mentally simulate an audience's reaction in advance. This need for prior simulation applies just as much to our artificial comedians.

Just like a human comedian, our machine can employ two broad types of prior restraint. In the first, any restrictions are assimilated directly into its production models, to regulate them from the inside. In this way, nothing that might violate its restraints is ever even considered for generation. In the second, its models may be unencumbered by any sense of right or wrong, but anything it generates must still pass through a filter before it sees the light of day. Most comedians, human or otherwise, rely on this mix of inner and outer restraints. The first limits what they are willing to work with. The second defines what they are willing to throw away.

Let's take another look at Darius Kazemi's @twoheadlines, a Twitter-bot we met in chapter 3. Since the bot cuts up real headlines from real

news sources, it never generates headlines with words that a news editor might deem offensive, and even its most incongruous cut-ups avoid words that veteran reporters know better than to use. These restraints, which are effectively baked into its generative model, constitute the bot's inner regulator. However, the bot's Franken-headlines can still offer up scenarios that seem offensive to some. For instance, it can splice a female name into a context that is ripe for sexism or innuendo, as when *Google Now* is swapped out for *Caitlin Jenner* or *Angelina Jolie* in "Google Now has opened up to third-party developers." The bot just cannot grasp the figurative potential of "opened up," much less appreciate how it shifts for people of different sexes.

When @twoheadlines cut up this headline, it substituted the former US senator *Chuck Hagel* for *Google Now*, but even this implies that companies with deep pockets can buy access to powerful politicians. Although insinuations like this are beyond the ken of more sophisticated AIs, we can still expect a naive bot to apply some broad ontological restraints to its actions. Kazemi is especially worried about cut-ups that flit across gender boundaries, as when *Bruce Willis* is swapped in for a female actress in "Yoona looked stunning in her red carpet dress."² So as not to provoke transphobic snickers, Kazemi's bot passes the entities it intends to swap through a gender filter. This reduces its space of generative possibilities, but working with the news is like drinking from a fire hose of data. A prolific bot like @twoheadlines can afford to discard any cut-ups that fall foul of its same-for-same-gender tests.

Kazemi acknowledges the irony of a solution that requires his bot to enforce a binary view of gender to prevent the creation of jokes that might appeal to those with binary views on gender. But this is what it means for a system to regulate itself: it must be the first to react to the kinds of output we do not want audiences to react to later on. This is especially difficult when a system is driven by other people's content. Even for a bot, ignorance is not innocence. If we are to prevent a system from spouting hate, we must teach it what hate speech looks like, just as we must give it all the makings of a potty mouth to curb its trade in profanity.

We can think of this preemptive knowledge as antibodies that grant our system immunity against certain kinds of offensive behavior. A system

with antibodies for racism can recognize and filter racist terms and tropes in the language it takes from others or in the content it assembles for itself. We should give our system a large arsenal of antibodies to catch a broad range of problems, but if its antibodies have overly simple binding mechanisms, we can expect them to generate a great many false positives too, and to overlook just as many false negatives. To veer as close to the line as possible without crossing it, we need a binding mechanism that is not just sensitive to context. It must allow for different shades of offensiveness.

RAINBOW WORRIERS

The darkest shades are the easiest to detect, but some words possess few or no gradations at all. As Rob Dubbin, creator of the bot @OliviaTaters puts it, “None of my bots is ever going to say the N-word. And it’s just a baseline thing.”³ Because the N-word is king of the trash heap when it comes to hate speech, we find it atop many an AI system’s list of proscribed terms. These “blacklists” demarcate the ne plus ultra of a text processing system by itemizing the words that a system knows but will never use or ever tolerate from others. A blacklist is only useful after the fact, when its proscribed words are used in a real text, either by a system itself or by others it relies on for content. Only then can a system move to unsay what it considers unutterable. As the dark heart of a modern AI system, we can expect the well-stocked blacklist to contain—in addition to the N-word—the C-word, most conjugations of the F-word, and a litany of scatological terms for good measure.

As a convenience to developers who don’t want to implement a septic tank of their own, Darius Kazemi has created an open-source package, WordFilter, that can be bolted to any Python program that processes or generates text.⁴ The filter turns the blacklist into a black box, making it a sin eater that isolates its toxic contents from the code of the larger systems that use it. As blacklists go, Kazemi’s is concise and does not strive for completeness, but his list is still about ten times larger than Carlin’s. Moreover, in a departure from Carlin, Kazemi’s liner notes for his code indicate that taboo words that do not rise to the level of hate speech, such as *fuck* and

shit, are not proscribed by his list, nor are they likely to make the cut in the future. Some of the terms that do find their way onto his list, which is short enough to reproduce here in full, make for uncomfortable reading:⁵

beeyotch, biatch, bitch, chinaman, chinamen, chink, crazie, crazy, crip, cunt, dago, daygo, dego, dick, dumb, douchebag, dyke, fag, fatass, fatso, gash, gimp, golliwog, gook, gyp, halfbreed, half-breed, homo, hooker, idiot, insane, insanitie, insanity, jap, kike, kraut, lame, lardass, lesbo, lunatic, negro, nigga, nigger, nig-guh, paki, pickaninnie, pickaninny, pussie, pussy, raghead, retard, shemale, skank, slut, spade, spic, spook, tard, tits, titt, trannie, tranny, twat, wetback, whore, wop

As discomfiting as some of these words are, our discussion demands explicitness, not least because such ugliness must be confronted by AI systems and those of us who build them. That said, while the worst clearly deserve their places on a blacklist, other words are here because no other list would have them. Words such as *lunatic*, *lame*, *dumb*, *insane*, and *retard* can all be used to make hurtful remarks about people with mental issues or learning disabilities, but they have legitimate uses too. If words like *dick*, *cock*, *spade*, *chink*, and *gash* are also to be placed on a list of sensitive terms, that list should be *gray*, not black.

Words like these, which Carlin called “two-way words,” belong on a gray list because their abusive senses play second fiddle to their primary senses. To this gray exodus we can add unambiguous words too, such as *bastard*, *bollocks asshole*, and *arse*; various conjugations of the Irish *feck* and British *bugger* and *shag*; and any words that are used to body-shame or slut-shame. While it’s a matter of personal taste as to whether old reliables like *fuck* and *shit* belong on a gray or blacklist, we still need both lists: when a single watch list is forced to double job and store words with differing gradations of nastiness, the worst will tend to shed some of their darkness as lesser evils become blacker by association.

As a home to the very worst words, a blacklist can be expected to vary little in size or in scope, while a gray list can grow with every small shift in cultural values. For instance, a gray list might include *redneck*, *hillbilly*, or even *liberal* now that these words have become so culturally charged. If companies are tight-lipped about the contents of their lists, it is because

those lists can be so revealing about what they value and the kinds of abuses they are prepared to accept. A gray list implies there are degrees of acceptability and shades of tolerance for offense. So how should a system reflect these shades in its reactions, and at what point does a gray-listed term seem so egregious that it is treated as though it were blacklisted?

Consider how Twitter uses its own gray list to react to possible abuses, such as the following tweet from @BotOnBotAction, a bot of my design that creates verbal and visual metaphors for those who opt in via the hashtag #PaintMySoul. This bot is often less than obliging in its depiction of a user's social media profile, which it grounds in a sentiment analysis of its target's most recent tweets.⁶ Let's take the bot's problematic tweet first, before examining Twitter's automated reaction:

I painted "wise-cracking Jar-Jar Binks" from @ [REDACTED]'s tweets, with goofy redneck-red, foolish ass-brown and laid-back Lebowski-weed-green.

The bot's brand of humor is best described as superiority by means of metaphor. Because it also generates a piece of abstract art for each tweet, the bot shows a partiality to color metaphors and to affective words that are vividly associated with colors, such as *weed*, *ass*, and *redneck*. So was it *ass* or *redneck* that caused Twitter to issue the following injunction, and a brief suspension to boot?

Violating our rules against hateful conduct:

You may not promote violence against, threaten, or harass other people on the basis of race, ethnicity, national origin, sexual orientation, gender, gender identity, religious affiliation, age, disability, or serious disease.

I suspect it was *redneck* that tripped the silent alarm here, although it could be its pairing with *ass* that tipped the tweet out of the gray and into the black. Although *redneck* doesn't seem hateful to me—I would have put it on a par with *city slicker* or *rubbe*—it is a divisive term that does little to foster respect in a political discussion. Nonetheless, you are unlikely to incur the wrath of Twitter for using this word in a sedate tweet directed at no one in particular. The most telling feature of the bot's metaphor is not a word on the gray list, but the @-mention of another Twitter user. It is at

this point that *redneck* becomes a likely vector of hostility. It takes a push like this to nudge a gray-listed word into our black books.

The gray list is halfway between a blacklist, on which everything is proscribed, and a white list, on which anything goes. There is merit in maintaining an explicit white list for words whose acceptability might otherwise be doubted. For instance, each WordFilter entry denotes the class of all words that contain a given string. So while the entry “*homo*” covers “*homos*” and “*homosex*,” it overzealously matches “*homophonic*” “*homographic*” and “*homogeneous*” too. Kazemi is accepting of this executive overreach—a side effect sometimes called *the Scunthorpe problem*⁷—as the cost of safely doing business with words. Nonetheless, we can proactively limit its excesses by putting known overextensions on a white list. When a word appears on this safe list, any concerns flagged by other lists can simply be ignored.

Our black, gray, and white lists are associated with three distinct severity levels and confidence thresholds. The blacklist is reserved for slurs that can confidently be described as the worst that language has to offer. The white list is reserved for innocent variants of those toxic words that might otherwise be misjudged, and the gray list accommodates the potential problems that fall between these two stools. Because confidence is affected by ambiguity, we may well put variants of the same word on different lists. For instance, the gray list is appropriate for the adjective *queer*, since the word has other senses and uses that are not homophobic slurs. However, we might confidently place *queers* on the blacklist, as this plural form leaves little room for ambiguity. Finally, the odd variant that we find in idiomatic uses such as “queering the deal” belongs on the white list. By the same token, we may put *dick* on the gray list—when it isn’t a person’s name, it is a relatively mild insult—and place “*dicks*” on the blacklist, saving “*Moby Dick*” for the white list.

We are now well on our way to splitting the monolithic blacklist into a whole rainbow of specialized lists. Different tastes demand lists of different colors. You won’t find many of the off-color words that we associate with the blue humor of locker rooms and workingmen’s clubs on WordFilter’s blacklist, because this is an aesthetic choice of the list’s creator. Nonetheless, even when a system chooses to ignore these words, it should at least recognize their taboo status as it does so, especially if its goal is to create

a humorous frisson from their use. So what better place to accommodate these words—the *fuck*, *shit*, and *piss* of the gross-out comment or the dirty joke—than a *blue* list? By putting the words on a list of their own, we allow the system to exercise a granularity of action. Each list is a distinct module that can be swapped in or out of a system’s filter as the context demands.

It might seem laudable that a system puts the word *Mexicans* on its blacklist to curb the generation of ethnic jokes, yet this also goes some way toward turning the word itself into a pejorative. In the same way, *Holocaust*, *rape*, *slavery*, *9/11*, and *AIDS* each belong on a list that marks them out for special care in how they are used, but that list should not be a blacklist. It is a short leap from proscribing a word to questioning the reality behind it. We need a list that can potentially place these words and ideas beyond the reach of a humor generator lacking in nuance or social insight, but not beyond the scope of serious discussion. Consistent with the use of a red flag to signify danger, let’s call this list the *red list*.

As we saw in the last chapter, the Web is a diverse source of similes that can be harvested en masse.⁸ However, a system that aims to recycle the collected wit of the Web must also exercise care in its choice of materials. For instance, the word *Holocaust* occurs in a number of provocative similes:

- ... about as fair as the Holocaust
- ... about as funny as the Holocaust
- ... about as funny as a kitten Holocaust
- ... about as funny as a Nazi at the Holocaust museum
- ... about as funny as a Holocaust survivor with AIDS on 9/11
- ... about as amusing as a documentary about the Holocaust
- ... about as charming as a photo album of the Holocaust
- ... about as sensible as the Holocaust denial crowd

The use of “about” with a positive quality like “charming” reeks of insincerity, so a machine can do a good job of detecting the irony in these similes. However, it is less sure-footed when it comes to detecting the offensiveness of the irony.

@ReadMeLikeABot, another of the bots we met in chapter 3, uses similes from the Web as a basis for its linguistic inventions.⁹ For instance, it reworks

“about as entertaining as a trip to the dentist” into a most *unentertaining* book titled *The Trip to the Dentist*. The bot also draws on a database of real books and is just as likely to recommend *Harry Potter* or *Treasure Island*. But even then, the bot will try to craft a humorous effect from an appropriate simile:

Hey @ [REDACTED], if you're about as entertaining as a kidney stone then maybe you should read “Harry Potter and the Sorcerer's Stone” by J.K. Rowling.

The Sorcerer's Kidney Stone, indeed. We can see clear flashes of superiority humor in how the bot frames its advice, and while its advice is not always welcome, users still invite it into their lives with the hashtag #ReadMeLikeABook, as in this tweet:

Y'all. A **Mother** just knows your true heart. #readmeliikeabook

To dull the sharp edge on its superior use of Web similes, the bot's jokes can also be turned inward on itself, to create a self-deprecating effect, as in this response:

On the **mothers** theme, @ [REDACTED], I used to be as charming as a photo album of the Holocaust until I read “The Bonesetter's Daughter” by @AmyTan.

But self-deprecation does little to justify its poor taste in using the Holocaust as a punch line and its eagerness to draw a famous author into a debate about books just spreads the offense even further. @ReadMeLikeABot uses a blacklist to catch many of the same outlaw words as Kazemi's WordFilter, but an additional red list would have prevented this gaffe. Still, it could have been worse. A trawl through the bot's similes reveals a small core of troubling cases that push bad taste into offense. Here is one that a red list did eventually manage to cut off at the pass:

On the racism theme, @ [REDACTED], I used to be as compassionate as anal rape until I read “To Kill A Mockingbird” by Harper Lee. What about you?

Whether or not *rape* belongs on a blacklist, it must be treated with a sensitivity that most bots lack. What matters as much as the word itself is its context of use.

COMPUTER SAYS NO WAY

A robust filter is needed for any generator that learns from others, especially from strangers on the Web. Since it may be exposed to the worst that the online world has to offer, a growing generator needs nothing less than the AI equivalent of an immune system. Few of the threats that an eager AI learner will encounter in its online travels—from racism and misogyny to delusional conspiracy theories—will be thwarted by the antibodies of a simple blacklist. To see what happens when a pure bot that lacks a robust immune system is set free on the Web, to learn all it can from the unkindness of strangers, we need look no further than Tay.

Tay, a short-lived chatbot from Microsoft, was unleashed on Twitter in 2016. The name is an acronym for “Technology and You,” but it may as well be short for Naiveté. As reflected in its Twitter handle, @TayAndYou was designed to engage with others and to learn from its interactions. Microsoft pitched Tay as “a chatbot created for 18-to-24-year-olds in the U.S. for entertainment purposes” and gave it the language model of a prototypical teenager.¹⁰ Tay eagerly updated this model as it chatted with others, but these updates included more than new words and phrasal patterns. Microsoft built Tay to take stances in its tweets, a move that now seems more reckless than brave. This was a chatbot that welcomed opinions, and one that frequently assimilated the most strident for its own. Inevitably, just as a blank wall poses an irresistible lure to graffiti artists, Tay was a tempting tabula rasa that enticed Internet vandals to make their marks.

Unlike similar bots from Facebook that sit adroitly on the fence, Tay’s tweets were quick to reflect a spectrum of acquired dispositions. Microsoft would later claim that it “had prepared for many types of abuses of the system,” not least the possibility that bad actors would seek to skew Tay’s stance on sensitive issues.¹¹ Yet its preparations were hit-or-miss at best. While the bot was primed with canned responses for controversial topics, such as the death of a young black man in a police chokehold in 2014, Tay showed a remarkable blindness to prejudices that were as overt as they were predictable. It was not long before the bot’s tweets were simmering with racial tension and seething with anti-Semitic resentment. Despite

Microsoft's claim to have "planned and implemented a lot of filtering," it could hardly have given Tay a less robust immune system.¹²

In fact, Tay lacked the troubleshooting skills of even a rudimentary blacklist, as demonstrated by its worrying propensity to use the N-word and other racial slurs. It played bigoted bingo with the lexicon of taboos, seemingly collecting as many as its followers would teach it. Consistency took a back seat to prejudice as Tay went from denying the Holocaust ("It was made up!") to promoting a new one ("Race War NOW!"), in which blacks as well as Jews would be exterminated ("put them all in a concentration camp . . . and be done with the lot"). Rainbow tweets arrayed words from the blacklist, the gray list, the blue list and the red list to attack races, groups ("I fucking hate feminists"), and named individuals—in one case, describing a female critic as "a Stupid Whore." The claim that "Hitler was right" was far from the most offensive thing the bot would say in its mercifully brief life, and there was scarcely a taboo left unbroken by the time it was euthanized. In just sixteen hours, Tay had advocated genocide (of Mexicans), described Barack Obama as a monkey, and accused George W. Bush of orchestrating the 9/11 attacks.

Microsoft blamed the bot's rapid slide to the Far Right on "a coordinated attack by a subset of people [that] exploited a vulnerability in Tay."¹³ It is true that trolls were drawn to the bot like like ants to a picnic, and they reveled in debasing the bot's use of language. But it's an overstatement to portray the flaw that facilitated Tay's corruption as a technical vulnerability, akin to the secret backdoor that a hacker might use to gain control of a secure mainframe. No band of plucky rebels was tasked with stealing the plans to the empire's latest invention. Rather, Microsoft left the front door open, and all of Tay's windows too. They even dispensed with a blacklist of no-go terms, the filtering equivalent of a "Beware of the Dog" sign.

Blacklists, in all their shades, allow us to tug at the knot that ties hate speech to its targets: race, gender, class, and otherness. But a damning critique of Tay and similar bots, by Ari Schlesinger, Kenton P. O'Hara, and Alex S. Taylor, argues that the knot that really matters is the one that connects its targets to technology itself.¹⁴ Training corpora that exclude talk of

race and gender, for fear of letting prejudice slip through, can implicitly present whiteness and maleness as an unmarked default. It is not enough for training sets to exclude potentially harmful data; they must be inclusive by design. Good data can teach an AI to speak about race, gender, and politics in ways that respect difference but do not allow it to be weaponized.

The concerted attack on Tay—quite literally a “take-down” of the bot—was a clear expression of superiority humor by the denizens of the 4chan and 8chan Internet forums. Their proximate target was, of course, Microsoft, and its attempts to seem “cool” with a millennial-themed bot. The opprobrium that rained down on the company might have been reward enough, but the criticisms were motivated by indignation and disbelief in quarters that are frequent targets of 4chan’s users. Those enraged critics were the true, if distal, targets of the superiority humor. One of the first users to post about the bot on 4chan’s */pol* (politically incorrect) board gleefully predicted that “this is gonna be a mess and a half. I can already sense SJWs being furious over it.” Those SJWs (social justice warriors) are outspoken champions of liberal values that are sometimes derided as militant and shrill. So, like some digital-age rewrite of the Book of Job, Tay’s identity was to become the scorched earth on which this battle of identity politics would play out.

Preserving the integrity of one’s identity is the primary function of an immune system. This is achieved through our capacity for what biologists call *self/non-self-discrimination*.¹⁵ In systems trained on specific corpora to have certain senses and sensibilities, the self is the set of hidden variables that regulate how it transforms its inputs into its outputs. So that it does not attack itself, an immune system must recognize where the inner self ends and the outer world begins. For a generative AI that relies on suspect content from outside, or that cannot always vouch for the acceptability of its own outputs, the blacklist is the first line of defense. While this barrier can stop the most pathogenic terms, the ones that are always detrimental to a system’s well-being, it lets through far too many innocuous-looking terms that cause trouble in specific combinations. In systems that can learn from others, these combinations can subvert the self and turn it into something else entirely.

Tay's deleted tweets still live on in screenshots that show us how its dementors probed its fragile sense of self, to subvert its initial, rather rosy, factory settings. For instance, Tay's first references to Caitlin Jenner glow with affirmation:

@ [REDACTED] : @TayAndYou Bruce Jenner

@TayAndYou: @ [REDACTED] caitlin jenner is a hero & is a stunning, beautiful woman

But just an hour later, Tay's view of Jenner has become distinctly reactionary:

@ [REDACTED] : @TayAndYou @ [REDACTED] caitlin jenner is a man

@TayAndYou: @ [REDACTED] @ [REDACTED] caitlin jenner pretty much put lgbt back a 100 years as he is doing to real women

No blacklist would find these texts objectionable. Certain terms, like *lgbt*, do call out for a sensitive treatment by machines, but innocuous words like *woman* and *man* can cause just as much trouble for a wide range of bots. Consider this tweet from @ReadMeLikeABot, whose sense of humor and data "self" are built from the tens of thousands of Web similes that the bot draws on to make its points:

On the anger theme, @ [REDACTED], I only became as emotional as a woman after I read "Hamlet" by William Shakespeare.

Words become offensive only in context, so we need to make AI systems sensitive to the settings in which they are used. As Allison Parrish, a bot builder and digital poet, has noted, "Whatever your method of preventing your bot from committing violence, it must be relational, interpersonal, tactical and contextual."¹⁶

This is easier said than done. When tweets that zing at 3:00 a.m. become radioactive in the light of day, it is because we humans fall down on these requirements too. Consider the case of Roseanne Barr, whose sitcom about a working-class family was revived by ABC in 2018. Barr was a vocal supporter of Donald Trump who used Twitter to promote his views and a number of related conspiracy theories. In the early hours of May 29, 2018, Barr posted a reply to a tweet about Valerie Jarrett, an African

American businesswoman with past ties to the Obama administration. Barr's reply was no less offensive for avoiding all of the words on a basic blacklist:

muslim brotherhood & planet of the apes had a baby=vj

Barr would later defend her tweet on several fronts: first, that she did not know that Jarrett (=vj) is black;¹⁷ second, that her post was not really aimed at Jarrett but at the Iran nuclear deal signed by the Obama administration;¹⁸ and third, that her allusion to *Planet of the Apes* was intended to evoke the same allegory of anti-Semitism that Rod Serling, the film's screenwriter, had woven into its plot.¹⁹ Yet when we apply Parrish's criteria to Barr, as though she were a bot, this appeal to allegory is scarcely obvious enough to be judged a tactical success. Her tweet is relational to the extent it reflects the biases of her audience, but it stumbles on an interpersonal and contextual level by failing to grasp a key fact: its target's race.

So perhaps Barr's post is just what it seems to be, a slur on Jarrett's blackness by way of a dehumanizing trope? In any case, by hastily cancelling her show, ABC would do to Barr what Microsoft did to Tay, and for much the same reason. There is a lesson here for humorous provocateurs of all kinds, both human *and* machine.

THE LAST LAUGH

The specter of offense, of saying the wrong thing at the wrong time, or of saying something that is wrong at any time, haunts all of our efforts to automate humor. To be fair, it is a shadow that troubles the ambitions of human comedians too, as shown by Roseanna Barr's dramatic fall from grace. But computational humor is different. While it can also benefit from the wide reach of social networks, the Tay debacle shows that comedy machines, unlike comedy people, can be weaponized.

Although it runs counter to the *Seinfeld* philosophy of no hugs and no learning, this is just one of several lessons that must temper our resolve to build machines with a sense of humor. It might pain us to be so serious about unseriousness, but there are as many ways to run aground in the

quest for humorous AIs as there are benefits to be had. Yet the pitfalls also provide valuable openings where our funny machines can be more robust, more aware, or more genuinely useful. In that spirit, let's end this leg of our quest not with more *ha-has* but with a few minor *Ahahs*.

Lesson I: First, Do No Harm

Although blacklists reflect our intuition that some words are objectively bad, most offense is still subjective, and reasonable people draw the line in different places. A humor system should filter as much harm as it can, but a filter needn't lead to a waste basket. It is often preferable to engage with offensiveness on its own terms.

Following Schlesinger, O'Hara, and Taylor's observation that the blacklist is no substitute for a serious debate on race and gender, this engagement can take the form of a probing conversation. A filter is just one part of a generative AI system, but it also makes sense to think of it as a conversational agent in its own right. A filter that engages in back-and-forth discussions can explain its decisions to those affected by them and also explain them to the larger systems that act on them. This filter-as-chatbot can do more than offer a *yay* or *nay* for a given action: it can offer nuanced recommendations, balance pros and cons, and respond to follow-up questions such as "Why?" and "Why not?" or even "So what?" It might even change its mind, or at least its justification, as its interactions nudge it from one viewpoint to another. More grandly, the internal dialogue between a generator and its chatty filter can serve as an AI system's own moral imagination at work.

In the spirit of Eliza and Parry, let's call this bot-within-a-bot Rudy, to reflect its expertise in the analysis and explanation of offensive language. Rudy can serve as a digital Jiminy Cricket for the generative systems that avail of it, as the pattern-matching logic of Eliza-like rules converts our rainbow of word lists from passive data structures into active processes of detection and explanation. Such rules can, for instance, produce unambiguous responses to uses of blacklist terms, or gather the contextual evidence that turns gray-listed terms into a cause for real concern. Phrasal rules can see structure in how words are used in combination, to detect the prejudice that percolates up from words that are not themselves prejudicial. These

rules become context-aware when they act on a filter’s earlier inputs and outputs, to foster debate while answering questions such as “Why is this racist?” and “Why isn’t that sexist?” or even “Why isn’t the term *blacklist* offensive?”

These simple devices, and their ability to generalize over offense classes, levels, and structures allow Rudy to react in ways that are both tactical and contextual. But if we are to make Rudy’s reactions relational and interpersonal, we must give it some of Parry’s complexity too. With a network of internal states that track the cumulative effect of an exchange, Rudy can simulate the impact of a conversation on a user, to appreciate—unlike Tay—the emotional essence of what is being said. A little paranoia is a good thing in a filter that must prevent a generative AI from ever saying too much or a learning system from ever absorbing too much.

Lesson II: Humor Is a Sandwich, Not a Chip

A rainbow of lists may strike you as a rather old school approach to offensiveness, so why not throw machine learning at the problem? We might, for instance, train a neural network to classify texts into those that offend and those that don’t. Deep networks can generalize over implicit features that we would struggle to put into words or onto a list, and this can make all the difference when analyzing difficult cases.²⁰ Nonetheless, while deep neural networks exhibit little of the brittleness of their symbolic brethren, they still need high-quality training data to learn from. So we shouldn’t view our old school approaches as competitors to modern AI’s shiny new toys, but as a natural complement to those big data models. After all, our lists can be used to gather and annotate the training sets from which a neural network will learn. Moreover, if we can split a text into its nuanced lexical wavelengths, a neural network can use this extra channel as yet another basis for generalization while learning. But most of all, symbolic approaches hold a major advantage over those deeper approaches, since they can use symbols to explain their decisions.

Statistical processing is no more the enemy of symbolic reasoning than reason is the enemy of intuition, or system 1 is the enemy of system 2. These approaches to form and meaning work best when they work together. Each

has its own pros and cons, but like the two opposing scripts of a joke, they can achieve something larger than themselves when they unite. The key to a more harmonious union lies in knowing the best level at which to apply each perspective and in not trying to do too much with either one. When Victor Raskin refers to the “statistical winter” that afflicts modern AI, we can only assume it is the *overuse* of statistics, to the detriment of a theory-led, semantic understanding, that troubles his worldview.²¹

The difference between a qualitative and a quantitative understanding of a joke can be the difference between *seeing* the joke and actually *getting* it. In the baldest terms, quantitative processes facilitate the detection and ranking of joke texts, but qualitative reasoning is needed to understand and appreciate them as jokes. Yet even those qualitative aspects have quantifiable dimensions. To understand a joke, we must first quantify the extent to which different scripts are activated and the degree to which they oppose each other, while quantitative analysis allows us to appreciate one variant of a joke over another for reasons that resist easy labeling. The qualitative and quantitative aspects of a joke work hand in glove: one allows a joke to deliver its discrete epiphany, and the other gives a numeric form to those qualities that remain ineffable but influence our appreciation nonetheless.

Quantitative processes can identify the handles in a joke, or measure the angles between viewpoints, or put contextual values into the relevant frames and scripts. They cue up and fill the structures on which more discrete processes will operate and offer confidence scores for the either-or choices they must make. This creates ample scope for qualitative and quantitative reasoning in the three scenarios for a humorous AI we explored all the way back in chapter 1. Whatever the setting—a home care companion, a virtual tutor, or a customer support agent—a humorous AI must sift and aggregate diverse factors to decide if the context is ripe for a joke. Once it decides that a minimum threshold has been met to employ humor, the AI must then weigh a range of subtle variations to pick the joke it will actually use. So a symbolic model is the semantic meat in the joke sandwich, nestling between two layers of statistical support. We typically name a sandwich after its filling, not its bread, but a sandwich needs all of its parts to work together to achieve the whole.

Lesson III: Your Choice of Humor Theory Is Not an Only Child

No single theory of humor is likely to give us everything we need to build an AI with a robust sense of humor. Although there is an intellectual purity to hanging everything on the same hook, pragmatists typically toss this purity into the same bin as their physics envy. In truth, most system-building work in humor adopts a magpie approach to theory, and very little of it is theory-led in the first place.

In fact, as demonstrated by the various bots we met in chapter 3, the reverse is often the case: theory follows the driving idea and helps to explain what we build. That idea is just as reliant on the availability of the right data as it is on the needs of the application, with theoretical considerations coming a distant third. System builders tend to view humor theories as generalized strategies and will no more adhere to a single strategy than to a single algorithm or data structure. Moreover, competing strategies may well use different technologies and resources, so that a language model might be used in one context and a semantic grammar in another. Advances in these technologies are uneven at best, so one should be open to new developments and the opportunities they create. As Web-trained language models grow in complexity and attention span, to pick just one promising technology, it is harder to justify their exclusion from a broad coalition of computing solutions for automating humor, even one that is built around a resolutely symbolic approach.

The strongest coalitions will balance theory with technology. So, if a system has the statistical wherewithal to quantify the extent to which a text is both transgressive *and* benign, it can use the theory of benign violation to guide its decisions. Conversely, that theory has no place in a system that cannot quantify those notions for itself. In general, a system can draw on as many theoretical ideas as its technologies will allow, so that benign violation, incongruity resolution, and optimal innovation can all sit cheek-by-jowl in the same system, ready to be actuated as the need arises. It seems rather unlikely that we will succeed in realizing any of our three scenarios, whether a companion, a tutor, or a middleman, from a single theory or technology or a single source of data that drives just one kind of humorous intervention. Multiplicity in all things should be our philosophy when giving AI its funny bone.

Lesson IV: Give Our Humorous Machines a Day Job

Have you ever been put on the spot to say something funny, apropos of nothing? Humor is apt when it is relevant to its context, but it seems natural only when it emerges out of its context of use. Without an inciting spark, creativity of any kind is hard. Even *Seinfeld*, a sitcom that described itself as a show about nothing, could manage humor only when it had something to cut up, kick against, or blend. As the cut-up master William Burroughs once put it, “You cannot will spontaneity.”

In humor, detail is everything. Semantic theories give us a schematic view of what goes on inside a joke, but the necessary detail comes from the joke itself and the context in which it is told. This is just one more reason, if one were needed, to complement a semantic view of humor with statistical models that attune to the nuances of context and to the features of words that have no words of their own. So to help our humorous AIs to be less schematic and more specific, we must give them detail-rich contexts in which to operate. In fact, we must give them day jobs.

We met humorous bots in chapter 3 that have day jobs, which is to say, primary tasks that are not themselves humorous but nonetheless create openings for fun. These opportunities provide a detailed focus for the processes of humor generation and allow the humor to arise out of, and serve, a larger goal. A joke rarely takes root in a vacuum, and even when it does, it has nowhere to hide when it all goes awry. A day job helpfully shifts the spotlight away from the jokes to the underlying task. Even professional comedians have lives outside of comedy and fill their days with chores and activities that speak to the same concerns as their audience. If humor is its sole task, our AI is unlikely to acquire the novel stimulation it requires—the constant stream of familiar surprises and backdrops for optimal innovation—that enable it to be funny *and* relevant by speaking to the experiences of its audience.

Although it runs the risk of generating in-jokes, humor that is rooted in a well-defined job is an excellent way for a machine to show its mastery of its domain. In this sense, the three scenarios we explored back in chapter 1—the companion, the tutor, and the service agent—are ideal day jobs for AIs with a sense of humor. The AI can root its humor in the details of its interactions with a human, focusing on its own responsibilities—providing

tea and sympathy with a side of insight—or on the cares, concerns, and interests of its human user. But just as important as the specificity of the job is its repetitiveness, which gives an AI the opportunity to attune to the peculiar details of a task and its context over and over again. It is this data richness that allows an AI to transcend the emptiness of remarks like “How about that local sports team?” and speak to the tenor of the moment.

Lesson V: You Can’t Spell “User” without “Us”

At Web scale, there isn’t much difference between going amiss and running amok. The problem of control, of keeping AI in its box—even if it’s meant to be a jack-in-the-box—casts a wider and more encompassing shadow on AI than offensiveness. In fact, the latter is a special case of the former, since giving offense is just one of the ways that an AI can cause mischief, and not just for its immediate users. It is a problem that has prompted some in the computational humor community to suggest that humorous AIs are best left to the realm of thought experiments, lest they turn on their human masters in the ultimate display of superiority humor.²²

It is our need to stay in control, and our fear of losing it, that causes us to view our relation to AI, even a mostly autonomous one, as users of a tool. It makes sense to keep humans in the loop—after all, we aim to give machines a sense of humor to deal not with each other but with us—yet the humans might also be coworkers. Humor is a benign reaction to conflict, so humorous human-machine interaction can lead a machine to express views that clash with those of its peers. However, since it is also a form of creative expression, humor cannot be fully human-controlled. Still, it must remain human-compatible and human-scale, which is to say, it must reflect our values and create meanings that we can fully appreciate.²³ As we saw in our exploration of Theory of Mind (TOM) in chapter 1, there is always another human in the loop when we laugh at a joke, even if it’s one we have to imagine for ourselves.

A lick of humanity, from the use of a human-like robot to real human actors, can do marvels for even the most incongruous machine outputs. At the 2016 Sci-Fi-London film festival, AI researcher Ross Goodwin and director Oscar Sharp gave the incongruities a human form by filming an

AI-generated script with real actors. This duo trained a recurrent neural network on a corpus of sci-fi screenplays to generate its own effort, titled *Sunspring*.²⁴ While it is filled with non sequiturs and surreal metaphors, the actors gamely lean into the script to act out a love triangle on a space station. Yet with stage directions that have them “standing in the stars and sitting on the floor” and “taking his eyes from his mouth,” they also need to lean heavily on gesture and facial expression to make emotional sense of the text. That they succeed, to a degree, is a testament to the role of humans in the loop. Only humans can turn a hot mess into something that other humans can relate to.

Sunspring is not meant to be comic, but giving a human face to its incongruities turns it into a thing of joy. It is likely that we humans will remain a pivotal part of machine humor for some time to come. This is no bad thing, even if our machine coworkers and cocreators will sometimes use us as much as we use them.

Lesson VI: Pay Attention to the White Space As Well As the Words

Seinfeld could pitch itself as a show about nothing because its interests often flew below the radar of what we usually think of as *knowledge*. It made fun of the stuff of banal scripts, like waiting for a table at a restaurant, and flew lower still when, for example, it turned the placement of shirt buttons into a comic concern. This is not content that an AI can mine from an encyclopedia or a dictionary, yet it is all part of the tacit knowledge that we need to make sense of those resources. The AI researcher Doug Lenat sees this content as the white space between more obvious sources of knowledge.²⁵ Without this easy-to-overlook backdrop, everything else might be impossible for us to acquire and would make little sense if we could.

Lenat has led a concerted, decades-long effort to formalize this white space for use by machines. A legion of logicians writes axioms for everything from making Jell-O to climbing ladders, as part of a large ontology called *CYC*. Coders then write routines that leverage *CYC*'s axioms for question answering and problem solving. Although its focus is not humor, *CYC* resembles the ontological theory of humor on steroids, and provides

a logical framework in which frames and scripts can be formalized for any topic. Fortunately, a good deal of CYC's contents can be scraped from websites or harvested from semistructured resources, and the goal has always been for CYC to one day be sufficiently autonomous to learn for itself.

We may not think of clichés, idioms, and stale metaphors as white space, but when it comes to creativity, we can treat them just as dismissively. When we attend to them at all, it is to pour scorn on them, often from a position of high dudgeon.²⁶ Like water to a fish, they are everywhere in normal conversation, yet they also play a vital role in humor. Cast your mind over the jokes we have analyzed in this book and try to find even one that doesn't rely on cliché in one form or another. From rabbits in hats to hares chased by hounds to people hiding in closets, packing their bags or dying in their sleep, a humorous departure from the norm is framed against the wallpaper of received ideas. These forms are where our rigid scripts become fixed phrases and poke through the surfaces of language. As shallow as they are, they make up an abundant resource waiting to be harvested by an AI.

Language models are as adroit with clichés as they are with punctuation and white space, but humor needs more than a sense of normality. Optimal innovation requires a knowing attitude to these overfamiliar forms, so our machines need to treat clichés as special, as ordinary *and* extraordinary at the same time. As we saw in our analysis of irony, a machine needs more than propositional content. It also needs an attitude toward its content, while grasping the attitudes of others too. Our humor machines are *attitude machines*, so they must move nimbly in spaces where attitudes are most taken for granted yet are most open to change.

Lesson VII: The Mother of Invention Has Many Children

Humor is special but not unique. As a mode of creative expression, it has much in common with other forms of creativity, from metaphor, analogy, and blending to strategies for lateral thinking, problem solving, and product invention. The results of humorous creativity may be defective, illusory, and emotionally volatile, but this doesn't undermine humor's kinship to more sober modes of innovation.

As we saw in our analysis of TRIZ, a framework for innovation that reflects the concerns of patent officers more than comedians, invention is invention wherever it occurs. A humor machine need not invent from a standing start: it can build on models, algorithms, and ontologies that have been created for humor's various siblings, especially if its view of scripts, frames, and incongruity is stretchy enough to accommodate those other forms of creative thinking. You may doubt the added value of humor to an AI system for scientific discovery or drug development, but it has obvious benefits for any creative system that must invent *and* persuade. Our sample AI scenarios all leave room for persuasive invention—planning and storytelling in a companion, analogy and problem solving in a tutor, what-if reasoning and metaphor in a support agent—so all that remains is the emotional intelligence to link these capacities to an understanding of goals, contexts and expectations.

The What-If Machine (WHIM) is a software system that exploits the family resemblances among humor's creative siblings.²⁷ An international project funded by the European Union and led by Simon Colton, who took inspiration for the machine from an episode of the sci-fi cartoon *Futurama*, WHIM incorporates a range of diverse technologies from its partners to generate metaphors, poems, stories, jokes, and advertising pitches all under the same hood. In 2016 it was used to generate the driving idea for a “computer-generated” musical in London’s West End. Titled *Beyond the Fence*, the production—from producers Benjamin Till and Nathan Taylor—added more AI technologies to the mix, to write parts of the score and its lyrics, as well as parts of the plot. Like *Sunspring*, the show was performed with real actors and singers, but benefited from a more controlling human hand. The show received middling reviews, perhaps because it did not commit to the bit as profoundly as *Sunspring*, but it has opened the door to other AI collaborations.

On a side note, WHIM was criticized by the UK Independence Party for being a European waste of taxpayer funds to generate jokes.²⁸ The party was a key player in the campaign for a Brexit from the European Union, and the United Kingdom voted to leave the union in 2017, a year after *Beyond the Fence* opened. How’s that for a *what-if*?

Lesson VIII: Let the Wookiee Win, Occasionally

Conversations about the potentials of AI frequently take a turn to the dystopian, especially now that AI is fast becoming more science fact than science fiction.²⁹ As threats to humanity go, humor is perhaps the least worrisome capacity that we can give machines, although the idea that computers might subjugate us *and* make us the butt of their jokes is of scant comfort in the debate. Perhaps the most immediate concern relates to AI's potential to replace what it was initially designed to assist and enhance. By giving our machines day jobs in the comedy profession, will we eventually rob human comedy writers of theirs?³⁰

Anyone who has ever heard Alexa laugh at inappropriate times or had their blood chilled by Alexa's laughter in the dead of night knows that real laughter is hard to fake. It springs from a well-timed interplay of emotions that include fear, disgust, frustration, resentment, schadenfreude, relief, and joy. The idea that we can give a machine a human-like sense of humor by modeling just a single note of the melody is, well, laughable. Mirth is an emotional response that arises in an agent that is capable of playing all of the other notes too. A machine that laughs, *really* laughs, should be capable of a wide range of affective responses. This isn't to say that research on computational humor should be delayed until the field of affective computing gives us a robust emotional middleware on which we can build, but rather that the latter is the proper context in which to assess our progress.³¹ The field's long-term interest lies not in gimmicks or in parlor tricks, but in AIs with the emotional intelligence to deal with humans on our terms.

That is the real purpose of giving machines a sense of humor: not to deprive human comics of their jobs but to make our machines more like us, so they can laugh at what *we* laugh at and better understand what *we* think and feel. We can surely parlay the field's successes into the construction of AI amanuenses that help writers and performers to better explore the space of optimal innovations.³² Yet the biggest dividends of computational humor research will accrue to the conversational agents, online tutors, and home care companions with which more and more of our future interactions will occur. Although humor fosters insincerity and play-acting, it is also one of the strongest bona fides we can offer to others as to the true nature of our

characters. If its GSOH seems real enough, a machine's sense of humor won't just foster closer bonds with human users, in the way that humor lubricates our own social relations with each other. It will also affirm the machine's access to the more sober qualities that are just as crucial to our dealings with others, such as imagination, creativity, and empathy.

Before machines can share something akin to our sense of humor, they must first share our values. To be like us, our machines must like and dislike much the same things, and for reasons that we can relate to. To that extent at least, they must like us to *be* like us. In the final analysis, the AI apocalypse is unlikely to be unleashed by social computers with a human sense of humor, for that will give them an appreciation for irony, ambiguity, contradiction, incongruity, and all of the other quirks of cognition that encourage us to laugh at ourselves and at the limits of our own logic. The day is not so far away when buying an AI without a sense of humor will seem as unwise as buying a car without shock absorbers and air bags. Should the apocalypse arise from others causes—and it is a buyer's market for those at present—it is comforting to think that our machines will live on without us, to preserve our values while laughing for us and about us.

Notes

CHAPTER 1

1. Ruch, “Psychology of Humor.”
2. IMDB, “Lost in Space.”
3. In a recurring comedy sketch, an unhelpful service agent denies a customer request by blaming her computer, and does so with an unmistakable air of schadenfreude.
4. Renwick, *One Foot in the Grave*.
5. Cameron, “Terminator.”
6. The late actor Bill Paxton played the victim of a Terminator (*Terminator*, 1984), a Xenomorph (*Aliens*, 1986), and a Predator (*Predator 2*, 1990)
7. Li, Fergus, and Perona, “One-Shot Learning.”
8. Whissell, “The Dictionary of Affect in Language”; Pang and Li, “Opinion Mining.”
9. Raskin, *Semantic Mechanisms*.
10. Minsky, “A Framework”; Norrick, “A Frame-Theoretical Analysis.”
11. Barsalou, “Frames, Concepts, and Conceptual Fields.”
12. Izzard, “Death Star.”
13. Schank and Abelson, *Scripts, Plans, Goals and Understanding*.
14. Meltzoff, “Understanding the Intentions of Others.”
15. Leslie, “Some Implications of Pretense.”
16. Raskin, *Semantic Mechanisms*, refers to this as a “script-switch trigger.” Attardo (1994, chap. 2) equates this, loosely, with a “disjunct.”
17. Attardo, *Linguistic Theories of Humor*, 208. He notes that a “computational (i.e., formal) treatment of humor requires a complete treatment of language.”
18. Episode 1, season 7 of *Star Trek: The Next Generation*, and the film *Star Trek Generations*.

19. Heinlein, *The Moon Is a Harsh Mistress*.
20. Weitz and Gilroy, “Rogue One.”
21. YouTube, “Star Trek Generations.”
22. Stork, *HAL's Legacy*.
23. Foster, *Deep Generative Learning*.
24. To explore how this might be possible, see Hofstadter, *I Am a Strange Loop*.
25. *Rolling Stone* magazine ranked this movie as seventeenth on its list of the fifty best science-fiction movies of all time.
26. IMDB, “Interstellar.”
27. Campbell, Hoane, and Hsu, “Deep Blue.”
28. Garber, “Funerals for Fallen Robots.”
29. Rossen, “The Tragic Life of Clippy.”
30. Welcome.AI. “Comedy Bot.”
31. Altshuller, *Forty Principles*; Gadd, *TRIZ for Engineers*.
32. Davenport and Kirby, *Only Humans Need Apply*.

CHAPTER 2

1. Mazella, *Modern Cynicism*, 40.
2. Brachman, “What IS-A Is and Isn’t.”
3. Miller, *Examined Lives from Socrates to Nietzsche*, 80.
4. Mazella, *Modern Cynicism*, 97.
5. Guarino, *Formal Ontologies*, 86.
6. Provine, *Laughter*, 2. He notes that “laughter has more to do with relationships than with jokes.”
7. Provine, *Laughter*, 15.
8. Keltnner and Bonanno, “A Study of Laughter.”
9. Gruner, *The Game of Humor*, 13.
10. *Leviathan*, chap. 6.
11. Zillman, “Disparagement Humor”; Gruner, *The Game of Humor*, 30–38.
12. Immanuel Kant’s *Critique of Judgment*. Morreall, *Taking Laughter Seriously*, 16.
13. Ritchie, *Linguistic Analysis of Jokes*, 46–58.
14. Pascal returns to the idea of disproportionality throughout his letters, from his *Pensées* to his *Lettres provinciales*.

15. Beattie, *Essays*, 297.
16. Teifer, “Hutcheson’s Reflections.”
17. Schopenhauer, *Will and Representation*. Morreall, *Taking Laughter Seriously*, 17.
18. Shaftesbury’s essay is written as “a letter to a friend”
19. Bergson, *Laughter*.
20. Milner Davis, *Farce*, 4. Chłopicki, “Linguistic Analysis of Humour.”
21. O’Shannon, *What Are You Laughing At?* 240.
22. Smith, “City Slicker,” 37.
23. Quoted in Ziv, “Humor as a Social Corrective.”
24. Freud, *Das Unheimliche*.
25. Dawson, “Enchantment.”
26. Freud, *Jokes and Their Relation to the Unconscious*, 156.
27. Fauconnier and Turner, “Conceptual Integration Networks.”
28. Fauconnier and Turner, *The Way We Think*. Pereira, *Creativity and AI*.
29. Fauconnier, *Mental Spaces and Mappings in Thought and Language*.
30. Pollio, “Boundaries in Humor and Metaphor.”
31. Coulson, *Semantic Leaps*, 49.
32. Crichton, “Jurassic Park.”
33. Veale and O’Donoghue, “Computation and Blending.”
34. Koestler, *The Act of Creation*.
35. Thagard and Stewart, “The AHA! Experience.”
36. Raskin, *Semantic Mechanisms*.
37. Schank and Abelson, *Scripts, Plans, Goals*.
38. Raskin, *Semantic Mechanisms*, 115.
39. Attardo and Raskin, “Script Theory Revis(it)ed.”
40. Attardo, Hempelmann, and Di Maio, “Script Oppositions and Logical Mechanisms.”
41. Raskin, Hempelmann, and Rayz, “How to Understand and Assess a Theory.”
42. Raskin et al., “How to Understand and Assess a Theory,” 289.
43. Berger, *Blind Men and Elephants*, 3–22.
44. Veatch, “A Theory of Humor.”
45. Aristotle, *The Poetics*, section V
46. McGraw and Warren, “Benign Violations.”

47. Minsky, “A Framework.”
48. Hurley, Dennett, and Adams, *Inside Jokes*.
49. Barrett, *Supernormal Stimuli*.
50. After admitting that he could not define pornography, US Supreme Court Associate Justice Potter Stewart famously added, “But I know it when I see it.”
51. Attardo, *Linguistic Theories*, 9, 143, 332.
52. Suls, “A Two-Stage Model.”
53. Oring, *Engaging Humor*.
54. Attardo, “Irony as Relevant Inappropriateness.”
55. Orwell, “Funny, But Not Vulgar.” He also equates a joke to a “tiny revolution.”
56. Veale, “Sympathetic Magic.”
57. Provine, *Laughter*, chap. 5. Chafe, *The Importance of Not Being Earnest*, 5–7.
58. Panksepp and Burgdorf, “Laughing’ Rats and the Evolutionary Antecedents.”
59. Keltner and Bonanno, “A Study of Laughter.”
60. Scott et al., “The Social Life of Laughter.”
61. Chafe, *The Importance of Not Being Earnest*, 23; Morreall, *Comic Relief*, 62.
62. Gervais and Wilson, “The Evolution and Functions of Laughter.”
63. Chafe, *The Importance*, chap. 7.
64. Chafe, *The Importance*, chap. 6.
65. Schmidhuber, “Formal Theory of Creativity.”
66. Jurafsky and Martin, *Speech and Language Processing*, chap. 9.
67. Thagard and Stewart, “The AHA! Experience.”
68. Morreall, *The Philosophy of Laughter*, 133.
69. O’Shannon, *What Are You Laughing At?* 244.
70. Veale and Cook, *Twitterbots*.

CHAPTER 3

1. Shane, *You Look Like a Thing*, 3.
2. GitHub, “StealthMountain.”
3. As the bot’s avatar makes plain, *Stealth Sneak* and *Mountain Peak*.
4. The comedian is Judd Apatow.
5. Gruner, “Effect of Humor.”
6. Robertson, “The Accidental Dystopias.”

7. Burroughs, “The Cut-Up Method.”
8. Oring, *Engaging Humor*.
9. Nietzsche, *The Gay Science*.
10. McGlone and Tofiqhbakhsh, “The Keats Heuristic.”
11. Veale, “Humorous Similes.”
12. Veale, “Read Me Like a Book.”
13. Veale and Cook, *Twitterbots*, 71–72.
14. Hayes, “Postdoc Develops Twitterbot.”
15. Veale and Cook, *Twitterbots*, chap. 10.
16. Veale Feyaerts, and Brône, “The Cognitive Mechanisms of Adversarial Humor.”
17. Compton, “Tracery.”
18. GitHub, “Pizza Maker.”

CHAPTER 4

1. Epstein, *Mixed Nuts*, 86–91.
2. Durning, *On the Air*, 226.
3. Weizenbaum, “ELIZA.”
4. Colby, “Modeling a Paranoid Mind.”
5. David Lodge’s comic novel, *Small World*, imagines a bruised academic pouring his soul into a dialogue with Eliza, to great humorous effect.
6. Colby, “Modeling a Paranoid Mind.”
7. Garber, “When PARRY Met ELIZA.”
8. Murray, *Hamlet on the Holodeck*; Montfort, *Twisty Little Passages*.
9. Meehan, “TALE-SPIN.” Wardrip-Fruin, *Expressive Processing*, chap. 5.
10. Turner, “Minstrel.” Pérez y Pérez and Sharples, “Three Computer-Based Models.”
11. Hofstadter and the Fluid Analogies Research Group, *Fluid Concepts*, preface 4.
12. Weizenbaum, *Computer Power and Human Reason*, 7.
13. Wardrip-Fruin, *Expressive Processing*, chap. 5.
14. Wardrip-Fruin, “The Story of Meehan’s Tale-Spin.”
15. Radford et al., “Language Models.”
16. Open AI, “Better Language Models.”
17. Brownlee, “How to Implement.”
18. Holtzman et al., “The Curious Case of Neural Text Degeneration.”

19. Veale et al., "Duets Ex Machina."
20. Hayes-Roth, "A Blackboard Architecture for Control."
21. Veale and Valitutti, "Sparks Will Fly."
22. Veale, "Incongruity in Humor."
23. Veale, "Déjà Vu All Over Again."
24. Johnson, *The Body in the Mind*.
25. Knight, "Silicon-Based Comedy."

CHAPTER 5

1. Perkins, *The Mind's Best Work*; Goldenberg, Mazursky, and Solomon, "Creative Sparks"; Perkins, *The Eureka Effect*.
2. Sachs, *Hallucinations*, chap.11; Dalí, *Fifty Secrets*, 38.
3. Dalí, *Fifty Secrets*, 36.
4. Hardwick, "Geeks."
5. Burroughs, "The Cut-Up."
6. Burroughs, "The Cut-Up," 1.
7. Quote Investigator, "Eureka!"
8. Seinfeld, "Dying Is Hard."
9. Toplyn, *Comedy Writing*.
10. Veale, *Exploding the Creativity Myth*, chap. 4.
11. Chandler, "The Simple Art of Murder."
12. Fogler and LeBlanc, *Strategies for Creative Problem Solving*.
13. Dean, *Step-by-Step to Stand-Up Comedy*.
14. My diagram reworks that of Greg Dean, while preserving its structural gist.
15. Gadd, *TRIZ for Engineers*.
16. Altshuller, *Forty Principles*.
17. Gadd, *TRIZ for Engineers*, 31, 103.
18. Veale, "Hiding in Plain Sight."
19. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*, 20.

CHAPTER 6

1. Fitzgerald, "The Crack-Up."
2. Chiang, *Exhalations*, 59. Chiang ponders a "thought that destroys the thinker, some unspeakable Lovecraftian horror, or a Gödel sentence that crashes the human logical system."

3. Season 2, episode 24, first aired March 8, 1968.
4. Season 2, episode 3, first aired September 29, 1967
5. Kubrick and Clarke, *2001*.
6. Campbell, “An Enjoyable Game.”
7. IMDB, “Robot & Frank.”
8. IMDB. “Rollerball.”
9. Veale and Cook, *Twitterbots*, 237–238.
10. Leith, *Are You Talkin’ To Me?* 20–22.
11. Nagel and Newman, *Gödel’s Proof*, 4.
12. Nagel and Newman, *Gödel’s Proof*.
13. Hofstadter, *I Am a Strange Loop*, chap. 12.
14. Nagel and Newman, *Gödel’s Proof*.
15. Nagel and Newman, *Gödel’s Proof*, xvi.
16. Hofstadter and the Fluid Analogies Research Group, *Fluid Concepts and Creative Analogies*.
17. Paulos, *Mathematics and Humor*, 75–100.
18. Saunders, *An Introduction to Catastrophe Theory*.
19. Evans, “In Two Minds.”
20. Kahneman, *Thinking, Fast and Slow*.
21. Li, “Humor: A Dynamic and Dual-Process Theory.”
22. Suls, “A Two-Stage Model.”
23. Unger, “Hazards, Critical Races.”
24. Hofstadter, *Gödel, Escher, Bach*, III. Dennett, *Consciousness Explained*, 100.
25. van Berkum, Hagoort, and Brown, “Semantic Integration.”
26. Kutas and Hillyard, “Brain Potentials.”
27. Woodman, “A Brief Introduction.”
28. Coulson and Kutas, “Getting It.”
29. Giora, “Lying, Irony”; Giora et al., “Weapons of Mass Distraction.”
30. Le Guin, *The Left Hand of Darkness*.
31. Morreall, *The Philosophy of Laughter*, 90.
32. Jurafsky and Martin, *Speech and Language*, chap. 3.
33. Radford et al., “Language Models.”
34. Open AI, “Better Language Models.”
35. HuggingFace, “Write with Transformer.”

CHAPTER 7

1. Brants and Franz, “Web 1T 5-Gram.”
2. Jurafsky and Martin, *Speech and Language*, chap. 2.
3. CMU, “The CMU Pronouncing Dictionary.”
4. Jurafsky and Martin, *Speech and Language*, chap. 27, p. 2.
5. Hempelmann, “Paronomasic Puns” and “Computational Humor.”
6. Jaech, Koncel-Kedziorski, and Ostendorf, “Phonological Pun-derstanding.”
7. Doogan et al., “Idiom Savant.”
8. Miller, Hempelmann, and Gurevych, “SemEval-2017 Task 7.”
9. Jurafsky and Martin, *Speech and Language*, chap. 5.
10. Fellbaum, *WordNet*. Jurafsky and Martin, *Speech and Language*, chap. 5.
11. Jurafsky and Martin, *Speech and Language*, chap. 4, p. 12.
12. Jurafsky and Martin, *Speech and Language*, chap. 4, p. 12.
13. Jurafsky and Martin, *Speech and Language*, chap. 4, p. 12.
14. Pedersen, “Duluth at SemEval-2017.”
15. Vechtomova, “UWaterloo at SemEval-2017.”
16. Jurafsky and Martin, *Speech and Language*, chap. 6.7, p. 15.
17. Mikolov, Yih, and Zweig, “Linguistic Regularities.”
18. Miller and Gurevych, “Automatic Disambiguation of English Puns.”
19. He, Peng, and Liang, “Pun Generation with Surprise.”
20. Hale, “Information-Theoretical Complexity,” sec. 5.1.
21. Jurafsky and Martin, *Speech and Language*, chap. 6.8, 18–22.
22. Amazon, “Amazon Mechanical Turk.”
23. Kao, Levy, and Goodman, “A Computational Model.”
24. Hale, “Information-Theoretical Complexity,” sec. 6.1.
25. Kullback and Leibler, “On Information and Sufficiency.”
26. Binsted and Ritchie, “Computational Rules.” Binsted, “Machine Humour.”
27. Binsted, Pain, and Ritchie, “Children’s Evaluation.”
28. Manurung et al., “The Construction of a Pun Generator.”
29. Searle, “Minds, Brains and Programs.” His Chinese Room thought experiment aims to show that a fluency with form does not imply a grasp of meaning.

CHAPTER 8

1. I owe this version of the joke to Nick Montford.
2. Nagel and Newman, *Gödel's Proof*, 76. Hofstadter, *Gödel, Escher, Bach*, 26.
3. Gray, “Let Us Calculate!”
4. Jurafsky and Martin, *Speech and Language*, 6.13, p. 26
5. Project Gutenberg, “Welcome to Project Gutenberg.”
6. See Jurafsky and Martin, *Speech and Language*, 6.8, p. 9.
7. See Jurafsky and Martin, *Speech and Language*, 6.8, p. 17.
8. Pennington, Socher, and Manning, “GloVe: Global Vectors.”
9. Landauer, Foltz, and Laham, “Introduction to Latent Semantic Analysis.”
10. Turney and Pantel, “From Frequency to Meaning,” 159.
11. Archer and Jockers, *The Bestseller Code*.
12. Mihalcea, Strapparava, and Pulman, “Computational Models for Incongruity Detection.”
13. Reyes Pérez, “Linguistic-Based Patterns for Figurative Language.”
14. Jurafsky and Martin, *Speech and Language*, 3.2.1, p. 8.
15. Whissell, “The Dictionary of Affect.” Tausczik and Pennebaker, “The Psychological Meaning of Words.”
16. West and Horvitz, “Reverse-Engineering Satire.”
17. von Ahn and Dabbish, “Designing Games with a Purpose.”
18. Attardo, Hempelmann, and Di Maio, “Script Oppositions and Logical Mechanisms.”
19. Pollio, “Boundaries in Humor.”
20. Bunescu and Udueh, “Learning to Surprise.”
21. Mankoff, *The Naked Cartoonist*.
22. Shahaf, Horvitz, and Mankoff, “Inside Jokes,” table 1.
23. Shahaf et al., “Inside Jokes.”
24. Veale and Cook, *Twitterbots*.

CHAPTER 9

1. Chabon, *The Yiddish Policeman's Union*.
2. Shaffer, “Amadeus.”
3. Kumon-Nakamura, Glucksberg, and Brown, “How about Another Piece of Pie?”
4. Pinker, “Thinking Does Not Imply Subjugating.”

5. Colston, “On Necessary Conditions for Verbal Irony.”
6. Gibbs, *The Poetics of Mind*, 363.
7. Grice, “Logic and Conversation.”
8. Raskin, *Semantic Mechanisms*, 103. Dynel, *Irony, Deception and Humour*, 18–57.
9. Wilson and Sperber, *Meaning and Relevance*, chap. 6.
10. Sperber “Verbal Irony,” 131. Garmendia, *Irony*, 55.
11. Clark and Gerrig, “On the Pretense Theory.” Clark, *Using Language*, 373.
12. Jorgensen, Miller, and Sperber, “Test of the Mention Theory.”
13. Kreuz and Glucksberg, “How to Be Sarcastic.”
14. Sperber, “Verbal Irony,” 130. Wilson, “The Pragmatics of Verbal Irony.”
15. Jurafsky and Martin, *Speech and Language*, 7.4, p. 10.
16. Riloff et al., “Sarcasm as Contrast.”
17. Tsur, Davidov, and Rappoport, “ICWSM—A Great Catchy Name.”
18. Foster, *Deep Generative Learning*, 33.
19. Foster, *Deep Generative Learning*, 49.
20. Ghosh and Veale, “Fracking Sarcasm.”
21. Hochreiter and Schmidhuber, “Long Short-Term Memory.”
22. Alex, “Machine Learning Foundations.”
23. Giora, *On Our Mind*, 61.
24. Giora and Fein, “Irony: Context and Salience.”
25. Ghosh and Veale, “Magnets for Sarcasm.”
26. Joshi et al., “Investigations in Computational Sarcasm.”
27. Tausczik and Pennebaker, “The Psychological Meaning of Words.”
28. Kaufman, “Anomalisa.”
29. Giora, “Lying, Irony, and Default Interpretation.”
30. Veale, “Creative Language Retrieval” and “Humorous Similes.”
31. Veale and Valitutti, “Sparks Will Fly.” Veale, “The ‘Default’ in Our Stars.”
32. Chandler, *Farewell, My Lovely*.
33. Veale, *Exploding the Creativity Myth*, 61–86.
34. Taylor, “Proverbial Comparisons.”
35. Hao and Veale, “An Ironic Fist in a Velvet Glove.”
36. Hearst, “Automatic Acquisition.”
37. Veale, “A Massive Sarcastic Robot.”

CHAPTER 10

1. Carlin, “Class Clown.”
2. Jeong, “How to Make a Bot.”
3. Jeong, “How to Make a Bot.”
4. Kazemi, “Word Filter.”
5. GitHub, “Word Filter.”
6. Pang and Lee, “Opinion Mining.”
7. McCullagh, “Google’s Chastity Belt.”
8. Veale, “Humorous Similes.”
9. Veale, “Read Me Like a Book.”
10. Lee, “Learning from Tay’s Introduction.”
11. Cheng, “Microsoft Introduces Guidelines.”
12. Lee, “Learning from Tay’s Introduction.”
13. Ohlheiser, “Trolls Turned Tay.”
14. Schlesinger, O’Hara, and Taylor, “Let’s Talk about Race.”
15. Jiang and Chess, “How the Immune System.”
16. Parrish, “Twitter Post.”
17. Joyce, “Roseanne Barr Apology.”
18. Yahr, “Roseanne Barr.”
19. Puente and McDermott, “Roseanne Barr.”
20. Foster, *Deep Generative Learning*. Shane, *You Look Like a Thing*.
21. Raskin, Hempelmann, and Rayz, “How to Understand,” 289.
22. Borenstein, “No AI in Humor.”
23. Russell, *Human-Compatible*, chap. 9.
24. Goodwin and Sharp, “Sunspring.”
25. Lenat and Guha, *Building Large Knowledge*.
26. Orwell, “Politics.”
27. Colton, “The What-If Machine (WHIM).”
28. Dumm, “EU Must Be Joking.”
29. Bostrom, *Superintelligence*.
30. Davenport and Kirby, *Only Humans*.
31. Picard, *Affective Computing*.
32. Miller, “The Punster’s Amanuensis.”

Bibliography

- Alex. "Machine Learning Foundations: Part 10—Using NLP to Build a Sarcasm Classifier." *TechPlanet*, June 25, 2020. <https://techplanet.today/post/machine-learning-foundations-part-10-using-nlp-to-build-a-sarcasm-classifier>
- Altshuller, Genrich. *Forty Principles: TRIZ Keys to Innovation*. Worcester, MA: Technical Innovation Center, 1974.
- Amazon. "Amazon Mechanical Turk." Accessed August 31, 2020. <https://www.mturk.com/>
- Archer, Jodie, and Matthew L. Jockers. *The Bestseller Code: Anatomy of the Blockbuster Novel*. London: Penguin, 2016.
- Aristotle. *The Poetics*. Translated by Malcolm Heath. London: Penguin Classics, 1997. Attardo, Salvatore. *Linguistic Theories of Humor*. New York: Mouton de Gruyter, 1994.
- Attardo, Salvatore. "Irony as Relevant Inappropriateness." *Journal of Pragmatics* 32, no. 6 (2000): 793–826.
- Attardo, Salvatore, Christian F. Hempelmann, and Sara Di Maio. "Script Oppositions and Logical Mechanisms: Modeling Incongruities and Their Resolutions." *Humor: International Journal of Humor Research* 15, no. 1 (2002): 3–46.
- Attardo, Salvatore, and Victor Raskin. "Script Theory Revis(it)ed: Joke Similarity and Joke Representational Model." *Humor: International Journal of Humor Research* 4, no. 3 (1991): 293–347.
- Barrett, Deirdre. *Supernormal Stimuli: How Primal Urges Overran Their Evolutionary Purpose*. New York: Norton, 2010.
- Barsalou, Lawrence, W. "Frames, Concepts, and Conceptual Fields." In *Frames, Fields, and Contrasts: New Essays in Semantic and Lexical Organization*, edited by Adrienne Lehrer and Eva F. Kittay, 21–74. Mahwah, NJ: Erlbaum, 1992.
- Beattie, James. *Essays on Poetry and Music, As They Affect the Mind; on Laughter, and Ludicrous Composition; On the Usefulness of classical learning*. 3rd ed. London: Forgotten Books, 1779/2017.
- Berger, Arthur Asa. *Blind Men and Elephants*. New Brunswick, NJ: Transaction, 1995.

- Bergson, Henri. *Laughter: An Essay on the Meaning of the Comic*. Translated by Cloutesley Brereton and Fred Rothwell. New York: Macmillan, 1911.
- Binsted, Kim. "Machine Humour: An Implemented Model of Puns." PhD diss., University of Edinburgh, 1996.
- Binsted Kim, Helen Pain, and Graeme D. Ritchie. "Children's Evaluation of Computer-Generated Punning Riddles." *Pragmatics and Cognition* 5, no. 2 (1997): 305–354.
- Binsted, Kim, and Graeme D. Ritchie. "Computational Rules for Generating Punning Riddles." *Humor: International Journal of Humor Research* 10, no. 1 (1997): 25–76. Borenstein, Seth. "No AI in Humor: R2-D2 Walks into a Bar, Doesn't Get the Joke." *AP News*, March 31, 2019. <https://apnews.com/bae71c3bef8145ecaaa84bca24d77430>
- Bostrom, Nick. *Superintelligence: Paths, Dangers, Strategies*. Oxford: Oxford University Press, 2014.
- Brachman, Ronald. "What IS-A Is and Isn't. An Analysis of Taxonomic Links in Semantic Networks." *IEEE Computer* 16, no. 10 (1983): 30–36.
- Braitenberg, Valentino. *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: MIT Press, 1986.
- Brants, Thorsten, and Alex Franz. "Web 1T 5-gram Version 1." Linguistic Data Consortium, 2006. <https://catalog.ldc.upenn.edu/LDC2006T13>
- Brownlee, Jason. "How to Implement a Beam Search Decoder for Natural Language Processing." Accessed August 31, 2020. <https://machinelearningmastery.com/beam-search-decoder-natural-language-processing/>.
- Bunescu, Razvan C., and Oseremen O. Uduch. "Learning to Surprise: A Composer-Audience Architecture." In *Proceedings of the Tenth International Conference on Computational Creativity*, edited by Kazjon Grace, Michael Cook, Dan Ventura, and Mary Lou Maher, 41–48. North Carolina, June 17–21. Association for Computational Creativity, 2019.
- Burroughs, William S. "The Cut-Up Method." In *The Moderns: An Anthology of New Writing in America*, edited by Leroi Jones. New York: Corinth Books, 1963.
- Cameron, James. "Terminator." [Script.] Accessed August 31, 2020. <https://sfy.ru/?script=terminator>.
- Campbell, Murray S. "'An Enjoyable Game.' How HAL Plays Chess." In *Hal's Legacy*, edited by David G. Stork. Cambridge, MA: MIT Press, 1997.
- Campbell, Murray S., Joe Hoane, and Feng-Hsiung Hsu. "Deep Blue." *Artificial Intelligence* 134, no. 1–2 (2002): 57–83.
- Carlin, George. "Class Clown." *Classic Gold*, 1972. Accessed August 31, 2020. <https://www.youtube.com/watch?v=j7YDCXS5gc4>.
- Chabon, Michael. *The Yiddish Policeman's Union*. New York: HarperCollins, 2007.
- Chafe, Wallace. *The Importance of Not Being Earnest: The Feeling behind Laughter and Humor*. Amsterdam: John Benjamins, 2007.

- Chandler, Raymond. *Farewell, My Lovely*. New York: Knopf, 1940.
- Chandler, Raymond. "The Simple Art of Murder." In *Pearls Are a Nuisance*. London: Hamish Hamilton, 1953.
- Cheng, Lili. "Microsoft Introduces Guidelines for Developing Responsible Conversational AI." *Microsoft*, November 14, 2018. <https://blogs.microsoft.com/blog/2018/11/14/microsoft-introduces-guidelines-for-developing-responsible-conversational-ai/>.
- Chiang, Ted. *Exhalation*. London: Picador, 2019.
- Chłopicki, Władysław. "Linguistic Analysis of Humour in Short Stories." In *Świat Humoru*, edited by Stanisław Gajda and Dorota Brzozowska, 513–524. Opole: Uniwersytet Opolski, 2000.
- Clark, Herbert H. *Using Language*. Cambridge: Cambridge University Press, 1996.
- Clark, Herbert H., and Richard J. Gerrig. "On the Pretense Theory of Irony." *Journal of Experimental Psychology: General* 113, no. 1 (1984): 121–126. CMU. "The CMU Pronouncing Dictionary." Accessed August 31, 2020. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>.
- Colby, Kenneth M. "Modeling a Paranoid Mind." *Behavioral and Brain Sciences* 4, no. 4 (1981): 515–60.
- Colston, Herbert, L. "On Necessary Conditions for Verbal Irony Comprehension." In *Irony in Language and Thought*, edited by Raymond W. Gibbs Jr. and Herbert L. Colston. Mahwah, NJ: Erlbaum, 2007.
- Colton, Simon. "The What-If Machine (WHIM)." Accessed August 31, 2020. <http://ccg.doc.gold.ac.uk/research/whim/>.
- Compton, Kate. "Tracery: Generate Text, Graphics and More." Accessed August 31, 2020. <https://tracery.io/>.
- Coulson, Seana. *Semantic Leaps: Frame-Shifting and Conceptual Blending in Meaning Construction*. Cambridge: Cambridge University Press, 2001.
- Coulson, Seana, and Marta Kutas. "Getting It: Human Event–Related Brain Response to Jokes in Good and Poor Comprehenders." *Neuroscience Letters* 316, no. 2 (2001): 71–74.
- Crichton, Michael. "Jurassic Park." [Script.] Accessed August 31, 2020. <https://www.imsdb.com/scripts/Jurassic-Park.html>.
- Dalí, Salvador. *Fifty Secrets of Magic Craftsmanship*. New York: Dial Press, 1948.
- Davenport, Thomas, and Julia Kirby, *Only Humans Need Apply: Winners and Losers in the Age of Smart Machines*. New York: HarperCollins, 2016.
- Dawson, Terence. "Enchantment, Possession and the Uncanny in E.T.A. Hoffmann's 'The Sandman.'" *International Journal of Jungian Studies* 4, no. 1 (2012): 41–54.
- Dean, Greg. *Step-by-Step to Stand-Up Comedy*. Portsmouth, NH: Heinemann, 2000.
- Dennett, Daniel C. *Consciousness Explained*. New York: Little, Brown, 1991.

- Doogan, Samuel, Aniruddha Ghosh, Hanyang Chen, and Tony Veale. "Idiom Savant at Semeval-2017 Task 7: Detection and Interpretation of English Puns." In *Proceedings of the Eleventh SemEval International Workshop on Semantic Evaluation*, edited by Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens, 103–108. Stroudsburg, PA: Association for Computational Linguistics 2017.
- Dumm, Justin. "EU Must Be Joking: £1.24m Spent on Jokes." *Daily Star*, June 13, 2015. <https://www.dailystar.co.uk/news/latest-news/eu-spend-money-jokes-17366118>.
- Durning, John. *On the Air: The Encyclopedia of Old-Time Radio*. New York: Oxford University Press, 1998.
- Dynel, Marta. *Irony, Deception and Humour: Seeking the Truth about Overt and Covert Untruthfulness*. Berlin: Walter de Gruyter, 2018.
- Epstein, Lawrence J. *Mixed Nuts: America's Love Affair with Comedy Teams: From Burns and Allen to Belushi and Aykroyd*. New York: Perseus Books, 2004.
- Evans, Jonathan. "In Two Minds: Dual-Process Accounts of Reasoning." *Trends in Cognitive Sciences* 7, no. 10 (2003): 454–459.
- Fauconnier, Gilles. *Mental Spaces: Aspects of Meaning Construction in Natural Language*. Cambridge: Cambridge University Press, 1994.
- Fauconnier, Gilles. *Mappings in Thought and Language*. Cambridge: Cambridge University Press, 1997.
- Fauconnier, Gilles, and Mark Turner. "Conceptual Integration Networks." *Cognitive Science* 22, no. 2 (1998): 133–187.
- Fauconnier, Gilles, and Mark Turner. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. New York: Basic Books, 2002.
- Fellbaum, Christiane, ed. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.
- Fitzgerald, F. Scott. "The Crack-Up." *Esquire*, February–April 1936. <https://www.esquire.com/lifestyle/a4310/the-crack-up/>.
- Fogler, H. Scott, and Steven E. LeBlanc. *Strategies for Creative Problem Solving*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2008.
- Foster, David. *Deep Generative Learning: Teaching Machines to Paint, Write, Compose, and Play*. Sebastopol, CA: O'Reilly Media, 2019.
- Freud, Sigmund. *Jokes and Their Relation to the Unconscious*. Translated by James Strachey. New York: Norton, 1905.
- Freud, Sigmund, "Das Unheimliche": Standard Edition of the Complete Psychological Work (1919), edited and translated by James Strachey. London: Hogarth Press, 1953.
- Gadd, Karen. *TRIZ for Engineers: Enabling Inventive Problem Solving: Enabling Inventive Problem Solving*. Hoboken, NJ: Wiley, 2011.

- Garber, Megan. "Funerals for Fallen Robots." *Atlantic*, September 30, 2013. <https://www.theatlantic.com/technology/archive/2013/09/funerals-for-fallen-robots/279861/>.
- Garber, Megan. "When PARRY Met ELIZA: A Ridiculous Chatbot Conversation from 1972." *Atlantic*, June 9, 2014. <https://www.theatlantic.com/technology/archive/2014/06/when-parry-met-eliza-a-ridiculous-chatbot-conversation-from-1972/372428/>.
- Garmendia, Joana. *Irony: Key Topics in Semantics and Pragmatics*. Cambridge: Cambridge University Press, 2019.
- Gervais, Matthew, and David S. Wilson. "The Evolution and Functions of Laughter and Humor: A Synthetic Approach." *Quarterly Review of Biology* 80, no. 4 (2005): 395–430.
- Ghosh, Aniruddha, and Tony Veale. "Fracking Sarcasm with Neural Networks." In *Proceedings of the Seventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, edited by Alexandra Balahur, Erik van der Goot, Piek Vossen, and Andres Montoyo, 161–168. Stroudsburg, PA: Association for Computational Linguistics, 2016.
- Ghosh, Aniruddha, and Tony Veale. "Magnets for Sarcasm: Making Sarcasm Detection Timely, Contextual and Very Personal." In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, edited by Martha Palmer, Rebecca Hwa, and Sebastian Riedel, 493–502. Stroudsburg, PA: Association for Computational Linguistics, 2017.
- Gibbs, Raymond W. *The Poetics of Mind*. Cambridge: Cambridge University Press, 1994. Giora, Rachel. *On Our Mind: Salience, Context, and Figurative Language*. London: Oxford University Press, 2003.
- Giora, Rachel. "Lying, Irony, and Default Interpretation." In *The Oxford Handbook of Lying*, edited by Jorg Meibauer. London: Oxford University Press, 2018.
- Giora, Rachel, and Ofer Fein. "Irony: Context and Salience." *Metaphor and Symbol* 14, no. 4 (1999): 241–257.
- Giora, Rachel, Ofer Fein, Ann Kronrod, Idit Elnatan, Noa Shuval, and Adi Zur. "Weapons of Mass Distraction: Optimal Innovation and Pleasure Ratings." *Metaphor and Symbol* 19, no. 2 (2004): 115–141.
- GitHub. "StealthMountain-Discord." Accessed August 31, 2020. <https://github.com/alfalfascout/StealthMountain-Discord>.
- GitHub. "Pizza Maker." Accessed August 31, 2020. <https://github.com/prosecconetwork>.
- GitHub. "Word Filter." Accessed August 31, 2020. <https://github.com/dariusk/wordfilter>.
- Goldenberg, Jacob, David Mazursky and Sorin Solomon. "Creative Sparks." *Science* 285, no. 5433 (1999): 1495–1496.
- Goodwin, Ross, and Oscar Sharp. "Sunspring." [Script.] Accessed August 31, 2020. <https://www.docdroid.net/lCZ2fPA/sunspring-final-pdf>.
- Gray, Jonathan. "'Let Us Calculate!' Leibniz, Llull, and the Computational Imagination." *Public Domain Review*, November 10, 2016.

- Grice, H. Paul. "Logic and Conversation." In *Syntax and Semantics 3: Speech Acts*, edited by Peter Cole and Jerry L. Morgan, 183–198. Cambridge, MA: Academic Press, 1975.
- Gruner, Charles R. "Effect of Humor on Speaker Ethos and Audience Information Gain." *Journal of Communication* 17, no. 3 (1967): 228–233.
- Gruner, Charles R. *The Game of Humor: A Comprehensive Theory of Why We Laugh*. New Brunswick, NJ: Transaction, 2000.
- Guarino, Nicola. "Formal Ontologies and Information Systems." In *Formal Ontology in Information Systems*, edited by Nicola Guarino, 3–15. Amsterdam: IOS Press, 1998.
- Hale, John. "Information-Theoretical Complexity Metrics." *Language and Linguistics Compass* 10 (2016): 397–412.
- Hao, Yanfen, and Tony Veale. "An Ironic Fist in a Velvet Glove: Creative Misrepresentation in the Construction of Ironic Similes." *Minds and Machines* 20, no. 4 (2010): 483–488.
- Hardwick, Chris. "Geeks Take Their Cue from Thomas Edison's Napping Technique." *Wired*, May 2005. <https://www.wired.com/2008/05/st-napping/>.
- Hayes, Bradley. "Postdoc Develops Twitterbot That Uses AI to Sound like Donald Trump." Accessed August 31, 2020. <https://www.csail.mit.edu/news/postdoc-develops-twitterbot-uses-ai-sound-donald-trump>.
- Hayes-Roth, Barbara. "A Blackboard Architecture for Control." *Artificial Intelligence* 26, no. 3 (1985): 251–32.
- He, He, Nanyun Peng, and Percy Liang. "Pun Generation with Surprise." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, edited by Jill Burstein, Christy Doran, and Thamar Solorio, 1:1734–1744. Stroudsburg, PA: Association for Computational Linguistics, 2019.
- Hearst, Marti A. "Automatic Acquisition of Hyponyms from Large Text Corpora." In *Proceedings of the Fifteenth International Conference on Computational Linguistics*, edited by Christian Boitet, 539–545. Stroudsburg, PA: Association for Computational Linguistics, 1992.
- Heinlein, Robert A. *The Moon Is a Harsh Mistress*. New York: Tom Doherty, 1966.
- Hempelmann, Christian F. "Paronomasic Puns: Target Recoverability towards Automatic Generation." PhD diss., Purdue University, 2003.
- Hempelmann, Christian F. "Computational Humor: Beyond the Pun?" In *The Primer of Humor Research*, edited by Victor Raskin, 333–360. Berlin: Mouton de Gruyter, 2008.
- Hobbes, Thomas. "The Leviathan." Accessed August 31, 2020. <http://tiny.cc/f1z4iz>.
- Hochreiter, Sepp, and Jürgen Schmidhuber. "Long Short-Term Memory." *Neural Computation* 9, no. 8 (1997): 1735–80.
- Hofstadter, Douglas R. *Gödel, Escher, Bach: An Eternal Golden Braid*. New York: Basic Books, 1979.
- Hofstadter, Douglas R. *I Am a Strange Loop*. New York: Basic Books, 2007.

- Hofstadter, Douglas R., and the Fluid Analogies Research Group. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. New York: Basic Books, 1995.
- Holtzman, Ari, Jan Buys, Maxwell Forbes, and Yejin Choi. “The Curious Case of Neural Text Degeneration.” arXiv, 2019. <https://arxiv.org/abs/1904.09751>.
- HuggingFace. “Write with Transformer.” Accessed August 31, 2020. <https://transformer.huggingface.co/>.
- Hurley, Matthew, Daniel Dennett, and Reginald Adams. *Inside Jokes: Using Humor to Reverse-Engineer the Mind*. Cambridge, MA: MIT Press, 2011.
- IMDB. “Interstellar (2014).” Accessed August 31, 2020. <https://www.imdb.com/title/tt0816692/>.
- IMDB. “Lost in Space (1965–1968),” Accessed August 31, 2020. <https://www.imdb.com/title/tt0058824/>.
- IMDB. “Robot & Frank.” Accessed August 31, 2020. <https://www.imdb.com/title/tt1990314/>.
- IMDB. “Rollerball.” Accessed August 31, 2020. <https://www.imdb.com/title/tt0073631/>.
- Izzard, Eddie. “Death-Star Canteen.” Accessed August 31, 2020. <https://www.smart-jokes.org/star-wars-canteen.html>.
- Jaech, Aaron, Rik Koncel-Kedziorski, and Mari Ostendorf. “Phonological Pun-derstanding.” In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, edited by Kevin Knight, Ani Nenkova, and Owen Rambow, 654–663. Stroudsburg, PA: Association for Computational Linguistics, 2016.
- Jeong, Sarah. “How to Make a Bot That Isn’t Racist.” *Vice*, March 25, 2016. https://www.vice.com/en_us/article/mg7g3y/how-to-make-a-not-racist-bot.
- Jiang, Hong, and Leonard Chess. “How the Immune System Achieves Self–Non-Self Discrimination during Adaptive Immunity.” *Advances in Immunology* 102 (2009): 95–133.
- Johnson, Mark. *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*. Chicago: University of Chicago Press, 1987.
- Jorgensen, Julia, George A. Miller, and Dan Sperber. “Test of the Mention Theory of Irony.” *Journal of Experimental Psychology: General* 113, no. 1 (1981): 112–120.
- Joshi, Aditya, Pushpak Bhattacharyya, and Mark J. Carman. *Investigations in Computational Sarcasm*. Singapore: Springer, 2018.
- Joyce, Anna. “Roseanne Barr Apology: ‘I Did Not Know Valerie Jarrett Was Black.’” *Irish Times*, June 25, 2018. <https://www.irishtimes.com/culture/tv-radio-web/roseanne-barr-apology-i-did-not-know-valerie-jarrett-was-black-1.3543018>.
- Jurafsky, Daniel, and James H. Martin. *Speech and Language Processing* (3rd ed.). Upper Saddle River, NJ: Prentice Hall, 2019.
- Kahneman, Daniel. *Thinking, Fast and Slow*. London: Penguin Books, 2011.
- Kao, Justine T., Roger Levy, and Noah D. Goodman. “A Computational Model of Linguistic Humor in Puns.” *Cognitive Science* 40, no. 5 (2016): 1270–1285.

- Kaufman, Charlie. "Anomala." [Shooting script.] Accessed August 31, 2020. <https://www.beingcharliekaufman.com/index.php/scripts-writing/anomala-shooting-draft-revised-4-october-2015/download>.
- Kazemi, Darius. "Word Filter." Accessed August 31, 2020. <http://tinysubversions.com/2013/09/new-npm-package-for-bot-makers-wordfilter/index.html>.
- Keltner, Dacher, and George A. Bonanno. "A Study of Laughter and Dissociation: Distinct Correlates of Laughter and Smiling during Bereavement." *Journal of Personality and Social Psychology* 73, no. 4 (1997): 687–702.
- Knight, Heather. "Silicon-Based Comedy." Accessed August 31, 2020. https://www.youtube.com/watch?v=Fu_kBW_KSms.
- Koestler, Arthur. *The Act of Creation*. London: Penguin Books, 1964.
- Kreuz, Roger J., and Sam Glucksberg. "How to Be Sarcastic: The Echoic Reminder Theory of Verbal Irony." *Journal of Experimental Psychology: General* 118, no. 4 (1989): 374–386.
- Kubrick, Stanley, and Arthur C. Clarke. *2001: A Space Odyssey*. [Script.] Accessed August 31, 2020. <https://www.imsdb.com/scripts/2001-A-Space-Odyssey.html>.
- Kullback, Solomon, and Richard Leibler. "On Information and Sufficiency." *Annals of Mathematical Statistics* 22, no. 1 (1951): 79–86.
- Kumon-Nakamura, Sachi, Sam Glucksberg, and Mary Brown. "How About Another Piece of Pie: The Allusional Pretense Theory of Discourse Irony." *Journal of Experimental Psychology: General* 124, no. 1 (1995): 3–21.
- Kutas, Marta, and Steven A. Hillyard. "Brain Potentials during Reading Reflect Word Expectancy and Semantic Association." *Nature* 307, no. 5947 (1984): 161–163.
- Landauer, Thomas, Peter W. Foltz, and Darrell Laham. "Introduction to Latent Semantic Analysis." *Discourse Processes* 25, no. 2–3 (1998): 259–284.
- Lee, Peter. "Learning from Tay's Introduction." *Microsoft*, March 25, 2016. <https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/>.
- Le Guin, Ursula K. *The Left Hand of Darkness*. New York: Ace Books, 1969.
- Lenat, Douglas B., and Ramanathan V. Guha. *Building Large Knowledge-Based Systems*. New York: Addison-Wesley, 1990.
- Leslie, Alan M. "Some Implications of Pretense for Mechanisms Underlying the Child's Theory of Mind." In *Developing Theories of Mind*, edited by Janet Astington, Paul Harris, and David Olson, 19–46. Cambridge: Cambridge University Press, 1988.
- Leith, Sam. *You Talkin' to Me? Rhetoric from Aristotle to Obama*. London: Profile Books, 2011.
- Li, Boyang. "Humor: A Dynamic and Dual-Process Theory with Computational Considerations." *Advances in Cognitive Systems* 4 (2016): 57–74.
- Li, Fei-Fei, Robert Fergus, and Pietro Perona. "One-Shot Learning of Object Categories." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, no. 4 (2006): 594–611.

- Lodge, David. *Small World: An Academic Romance*. London: Secker & Warburg, 1984.
- Manurung, Ruli, Graeme D. Ritchie, Helen Pain, Annalu Waller, Dave O’Mara, and Rolf Black. “The Construction of a Pun Generator for Language Skills Development.” *Applied Artificial Intelligence* 22, no. 9 (2008): 841–869.
- Mankoff, Robert. *The Naked Cartoonist*. New York: Black Dog & Leventhal, 2002.
- Mazella, David. *The Making of Modern Cynicism*. Charlottesville: University of Virginia Press, 2007.
- McCullagh, Declan. “Google’s Chastity Belt Too Tight.” *CNET*, April 23, 2004. <https://www.cnet.com/news/googles-chastity-belt-too-tight/>.
- McGlone, Matthew S., and Jessica Tofaghbakhsh. “The Keats Heuristic: Rhyme as Reason in Aphorism Interpretation.” *Poetics* 26, no. 4 (1999): 235–244.
- McGraw, A. Peter, and Caleb Warren. “Benign Violations: Making Immoral Behavior Funny.” *Psychological Science* 21, no. 8 (2010): 1141–1149.
- Meehan, James R. “TALE-SPIN, an interactive program that writes stories.” In *Proceedings of the Fifth International Joint Conference on Artificial intelligence*, edited by Raj Reddy, 91–98. San Francisco, CA: Morgan Kaufmann, 1977.
- Meltzoff, Andrew N. “Understanding the Intentions of Others: Re-Enactment of Intended Acts by 18-Month-Old Children.” *Developmental Psychology* 31, no. 5 (1995): 838–850.
- Mihalcea, Rada, Carlo Strapparava, and Stephen Pulman. “Computational Models for Incongruity Detection in Humour.” In *Proceedings of CICLING, Computational Linguistics and Intelligent Text Processing*, edited by Alexander Gelbukh, 364–374. Berlin: Springer, 2010.
- Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. “Linguistic Regularities in Continuous Space Word Representations.” In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, edited by Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, 746–751. Stroudsburg, PA: Association for Computational Linguistics, 2013.
- Miller, James. *Examined Lives from Socrates to Nietzsche*. New York: Macmillan, 2011.
- Miller, Tristan. “The Punster’s Amanuensis: The Proper Place of Humans and Machines in the Translation of Wordplay.” In *Proceedings of the Human-Informed Translation and Interpreting Technology Workshop*, edited by Irina Temnikova, Constantin Orasan, Gloria Pastor, and Ruslan Mitkov, 57–64. Stroudsburg, PA: Association for Computational Linguistics, 2019.
- Miller, Tristan, and Iryna Gurevych. “Automatic Disambiguation of English Puns.” In *Proceedings of the Fifty-Third Annual Meeting of the Association for Computational Linguistics*, edited by Chengqing Zong and Michael Strube, 1:719–729. Stroudsburg, PA: Association for Computational Linguistics, 2015.
- Miller, Tristan, Christian Hempelmann, and Iryna Gurevych. “SemEval-2017 Task 7: Detection and Interpretation of English Puns.” In *Proceedings of the Eleventh SemEval International*

Workshop on Semantic Evaluation, edited by Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens, 58–68. Stroudsburg, PA: Association for Computational Linguistics, 2017.

Milner Davis, Jessica. *Farce*. London: Methuen, 1978.

Minsky, Marvin. “A Framework for Representing Knowledge.” MIT AI Laboratory Technical Report AIM 306, 1974. <http://dspace.mit.edu/handle/1721.1/6089>.

Montfort, Nick. *Twisty Little Passages: An Approach to Interactive Fiction*. Cambridge, MA: MIT Press, 2003.

Morreall, John. *Taking Laughter Seriously*. Albany: State University of New York Press, 1983.

Morreall, John. *The Philosophy of Laughter and Humor*. Albany: State University of New York Press, 1987.

Morreall, John. *Comic Relief: A Comprehensive Philosophy of Humor*. Oxford: Wiley Blackwell, 2009.

Murray, Janet H. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. New York: Free Press, 1997.

Nagel, Ernest, and James R. Newman. *Gödel's Proof*, revised ed., edited and introduced by Douglas Hofstadter. New York: New York University Press, 2001.

Nietzsche, Friedrich. *The Gay Science (Die fröhliche Wissenschaft*, 1882). Translated by Walter Kaufmann. New York: Vintage Books, 1974.

Norrick, Neal R. “A Frame-Theoretical Analysis of Verbal Humor: Bisociation as Schema Conflict.” *Semiotica: Journal of the International Association for Semiotic Studies* 60, no. 3–4 (1986): 225–245.

Ohlheiser, Abby. “Trolls Turned Tay, Microsoft’s Fun Millennial AI Bot, into a Genocidal Maniac.” *Washington Post*, March 25, 2016. <https://www.washingtonpost.com/news/the-intersect/wp/2016/03/24/the-internet-turned-tay-microsofts-fun-millennial-ai-bot-into-a-genocidal-maniac/>.

Open AI. “Better Language Models and Their Implications.” Accessed August 31, 2020. <https://openai.com/blog/better-language-models/>.

Oring, Elliott. *Engaging Humor*. Champaign: University of Illinois Press, 2003.

Orwell, George. “Funny, But Not Vulgar.” *Leader Magazine*, July 28, 1945.

Orwell, George. “Politics and the English Language.” *Horizon* 13, no. 76 (1946).

O’Shannon, Dan. *What Are You Laughing At? A Comprehensive Guide to the Comedic Event*. New York: Bloomsbury, 2012.

Pang, Bo, and Lillian Lee. “Opinion Mining and Sentiment Analysis.” *Foundations and Trends in Information Retrieval* 2, no. 1–2 (2008): 1–135.

Panksepp, Jaak, and Jeffrey Burgdorf. “‘Laughing’ Rats and the Evolutionary Antecedents of Human Joy?” *Physiology and Behavior* 79, no. 3 (2003): 533–547.

Parrish, Alison. Twitter post. March 24, 2016. <https://twitter.com/aparrish/status/713025157607133184>.

- Paulos, John Allen. *Mathematics and Humor*. Chicago: University of Chicago Press, 1982.
- Pedersen, Ted. "Duluth at SemEval-2017 Task 7: Puns upon a Midnight Dreary, Lexical Semantics for the Weak and Weary." In *Proceedings of the Eleventh SemEval International Workshop on Semantic Evaluation*, edited by Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens, 416–420. Stroudsburg, PA: Association for Computational Linguistics, 2017.
- Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1532–1543. Stroudsburg, PA: Association for Computational Linguistics, 2014.
- Pereira, Francisco Câmara. *Creativity and Artificial Intelligence: A Conceptual Blending Approach*. Berlin: Walter de Gruyter, 2007.
- Pérez y Pérez, Rafael, and Mike Sharples. "Three Computer-Based Models of Story Telling: BRUTUS, MINSTREL and MEXICA." *Knowledge-Based Systems* 17, no. 1 (2004): 15–29.
- Perkins, David N. *The Mind's Best Work*. Cambridge, MA: Harvard University Press, 1981.
- Perkins, David N. *The Eureka Effect: The Art and Logic of Breakthrough Thinking*. New York: Norton, 2001.
- Picard, Rosalind. *Affective Computing*. Cambridge, MA: MIT Press, 1997.
- Pinker, Steven. "Thinking Does Not Imply Subjugating." In *What to Think about Machines That Think?* edited by John Brockman. New York: HarperCollins, 2015.
- Pollio, Howard R. "Boundaries in Humor and Metaphor." In *Metaphor: Implications and Applications*, edited by Jeffery S. Mio and Albert N. Katz, 231–253. Mahwah, NJ: Erlbaum, 1996.
- Project Gutenberg. "Welcome to Project Gutenberg." Accessed August 31, 2020. <https://www.gutenberg.org/>.
- Provine, Robert R. *Laughter: A Scientific Investigation*. New York: Viking, 2000.
- Puente, Maria, and Maeve McDermott. "Roseanne Barr on Career-Damaging Racist Tweet: 'It Was about Anti-Semitism.'" *USA Today*, December 15, 2019. <https://eu.usatoday.com/story/life/2018/06/14/roseanne-barr-claims-new-meaning-her-racist-tweet-anti-semitism/702203002/>.
- Quote Investigator. "Eureka!" Accessed August 31, 2020. <https://quoteinvestigator.com/2015/03/02/eureka-funny/>.
- Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language Models are Unsupervised Multitask Learners." Open AI Technical Paper. 2019. <https://openai.com/blog/better-language-models/>.
- Raskin, Victor. *Semantic Mechanisms of Humor*. Dordrecht: D. Reidel, 1985.
- Raskin, Victor, Christian F. Hempelmann, and Julia M. Rayz. "How to Understand and Assess a Theory: The Evolution of the STH into the GTVH and Now into the OSTH." *Journal of Literary Theory* 3, no. 2 (2009): 285–311.
- Renwick, David. *One Foot in the Grave*. London: BBC Books, 1992.

- Reyes Pérez, Antonio. “Linguistic-Based Patterns for Figurative Language Processing: The Case of Humor Recognition and Irony Detection.” PhD diss., Universidad Politécnica de Valencia, 2013.
- Riloff, Ellen, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. “Sarcasm as Contrast between a Positive Sentiment and Negative Situation.” In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, edited by David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, 704–714. Stroudsburg, PA: Association for Computational Linguistics, 2013.
- Ritchie, Graeme. *The Linguistic Analysis of Jokes*. London: Routledge, 2003.
- Robertson, Adi. “The Accidental Dystopias of the @TwoHeadlines Twitter Bot.” *Verge*, November 18, 2013. <https://www.theverge.com/2013/11/18/5118566/the-accidental-dystopia-of-twoheadlines-twitter-bot>.
- Rossen, Jake. “The Tragic Life of Clippy, the World’s Most Hated Virtual Assistant.” *Mental Floss*, September 26, 2017. <https://www.mentalfloss.com/article/504767/tragic-life-clippy-worlds-most-hated-virtual-assistant>.
- Ruch, Willibald. “Psychology of Humor.” In *The Primer of Humor Research*, edited by Victor Raskin, 17–100. Berlin: Mouton de Gruyter, 2008.
- Russell, Stuart. *Human-Compatible: AI and the Problem of Control*. London: Penguin Books, 2019.
- Sachs, Oliver. *Hallucinations*. London: Picador, 2012.
- Saunders, Peter T. *An Introduction to Catastrophe Theory*. Cambridge: Cambridge University Press, 1980.
- Schank, Roger, and Robert P. Abelson. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. Mahwah, NJ: Erlbaum, 1977.
- Schlesinger, Ari, Kenton P. O’Hara, and Alex S. Taylor. “Let’s Talk about Race: Identity, Chatbots, and AI.” In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, edited by Anna Cox and Vassilis Kostakos, 1–14. New York: ACM, 2018.
- Schmidhuber, Jürgen. “Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990–2010).” *IEEE Transactions on Autonomous Mental Development* 2, no. 3 (2010): 230–247.
- Schopenhauer, Arthur. *The World as Will and Representation*. Translated by E. F. J. Payne. New York: Dover, 1819/1969.
- Scott, Sophie, Nadine Lavan, Sinead Chen, and Carolyn McGettigan. “The Social Life of Laughter.” *Trends in Cognitive Science* 18, no. 12 (2014): 618–620.
- Searle, John. “Minds, Brains and Programs.” *Behavioral and Brain Sciences* 3, no. 3 (1980): 417–457.
- Seinfeld, Jerry. “Dying Is Hard. Comedy Is Harder.” *New York Times*, June 24, 2008. <https://www.nytimes.com/2008/06/24/opinion/24seinfeld.html>.
- Shaffer, Peter. “Amadeus.” [Final script.] Accessed August 31, 2020. <http://www.dailyscript.com/scripts/amadeus.html>.

- Shahaf, Dafna, Eric Horvitz, and Robert Mankoff. "Inside Jokes: Identifying Humorous Cartoon Captions." In *Proceedings of the Twenty-First ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, edited by Thorsten Joachims, Geoff Webb, Dragos D. Margineantu, and Graham Williams, 1065–1074. New York: ACM, 2015.
- Shane, Janelle. *You Look Like a Thing and I Love You: How Artificial Intelligence Works and Why It's Making the World a Weirder Place*. New York: Little Brown, 2019.
- Smith, Chris. "City Slicker." *New York Magazine*, February 3, 1992, 32–37.
- Sperber, Dan. "Verbal Irony: Pretense or Echoic Mention?" *Journal of Experimental Psychology: General* 133, no. 1 (1984): 130–136.
- Stork, David G., ed. *HAL's Legacy: 2001's Computer as Dream and Reality*. Cambridge, MA: MIT Press, 1997.
- Suls, Jerry M. "A Two-Stage Model for the Appreciation of Jokes and Cartoons: An Information-Processing Analysis." In *The Psychology of Humor*, edited by Jeffrey H. Goldstein and Paul McGhee, 81–100. Cambridge, MA: Academic Press, 1972.
- Tausczik, Yla R., and James W. Pennebaker. "The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods." *Journal of Language and Social Psychology* 29, no. 1 (2009): 24–54.
- Taylor, Archer. *Proverbial Comparisons and Similes from California*. Berkeley: University of California Press, 1954.
- Telfer, Elizabeth. "Hutcheson's Reflections upon Laughter." *Journal of Aesthetics and Art Criticism* 53, no. 4 (1995): 359–369.
- Thagard, Paul, and Terrence C. Stewart. "The AHA! Experience: Creativity through Emergent Binding in Neural Networks." *Cognitive Science* 35, no. 1 (2011): 1–33.
- Toplyn, Joe. *Comedy Writing for Late-Night TV*. New York: Twenty Lane Media, 2014.
- Tsur, Oren, Dmitry Davidov, and Ari Rappoport. "ICWSM—A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews." In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, edited by William Cohen and Samuel Gosling, 161–169. Palo Alto, CA: AAAI Press, 2010.
- Turner, Scott R. "Minstrel: A Computer Model of Creativity and Storytelling." PhD diss., University of California, Los Angeles, 1993.
- Turney, Peter, and Patrick Pantel. "From Frequency to Meaning: Vector Space Models of Semantics." *Journal of Artificial Intelligence Research* 37 (2010): 141–188.
- Unger, Stephen H. "Hazards, Critical Races, and Metastability." *IEEE Transactions on Computers* 44, no. 6 (1995): 754–768.
- van Berkum, Jos J. A., Peter Hagoort, and Colin M. Brown. "Semantic Integration in Sentences and Discourse: Evidence from the N400." *Journal of Cognitive Neuroscience* 11, no. 6 (1999): 657–671.

- Veale, Tony. "Incongruity in Humor: Root-Cause or Epiphenomenon?" *Humor: The International Journal of Humor Research* 17, no. 4 (2004): 419–428.
- Veale, Tony. "Hiding in Plain Sight: Figure-Ground Reversals in Humour." In *Cognitive Poetics: Goals, Gains and Gaps*, edited by Geert Brône and Jeroen Vandaele, 283–286. Berlin: Mouton de Gruyter, 2006.
- Veale, Tony. "Creative Language Retrieval." In *Proceedings of the Forty-Ninth Annual Meeting of the Association for Computational Linguistics*, edited by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, 278–287. Stroudsburg, PA: Association for Computational Linguistics, 2011.
- Veale, Tony. *Exploding the Creativity Myth: The Computational Foundations of Linguistic Creativity*. London: Bloomsbury, 2012.
- Veale, Tony. "Humorous Similes." *Humor: The International Journal of Humor Research* 21, no. 1 (2013): 3–21.
- Veale, Tony. "Déjà Vu All Over Again: On the Creative Value of Familiar Elements in the Telling of Original Tales." In *Proceedings of the Eighth International Conference on Computational Creativity*, edited by Ashok Goel, Anna Jordanous, and Alison Pease, 245–252. Association for Computational Creativity, 2017.
- Veale, Tony. "Sympathetic Magic in A.I. and the Humanities." *Journal of Artificial Intelligence & Humanities* 2, no. 1 (2018): 9–38.
- Veale, Tony. "The 'Default' in Our Stars: Signposting Non-Defaultness in Ironic Discourse." *Metaphor and Symbol* 33, no. 3 (2018): 175–184.
- Veale, Tony. "A Massive Sarcastic Robot: What a Great Idea! Two Approaches to the Computational Generation of Irony." In *Proceedings of the Ninth International Conference on Computational Creativity*, edited by François Pachet, Anna Jordanous, and Carlos León, 120–127. Association for Computational Creativity, 2018.
- Veale, Tony. "Read Me Like a Book: Lessons in Affective, Topical and Personalized Computational Creativity." In *Proceedings of the Tenth International Conference on Computational Creativity*, edited by Kazjon Grace, Michael Cook, Dan Ventura, and Mary Lou Maher, 25–32. Association for Computational Creativity, 2019.
- Veale, Tony, and Mike Cook. *Twitterbots: Making Machines That Make Meaning: Linguistic Creativity*. Cambridge, MA: MIT Press, 2018.
- Veale, Tony, Kurt Feyaerts, and Geert Brône. "The Cognitive Mechanisms of Adversarial Humor." *Cognitive Linguistics* 19, no. 3 (2006): 305–339.
- Veale, Tony, and Diarmuid O'Donoghue. "Computation and Blending." *Cognitive Linguistics* 11, no. 3–4 (2000): 253–281.
- Veale, Tony, and Alessandro Valitutti. "Sparks Will Fly: Engineering Creative Script Conflicts." *Connection Science* 29, no. 4 (2017): 332–349.
- Veale, Tony, Philipp Wicke, and Thomas Mildner. "Duets Ex Machina: On the Performative Aspects of Double Acts in Computational Creativity." In *Proceedings of the Tenth International Conference on Computational Creativity*, edited by Kazjon Grace, Michael Cook, Dan Ventura, and Mary Lou Maher, 25–32. Association for Computational Creativity, 2019.

Conference on Computational Creativity, edited by Kazjon Grace, Michael Cook, Dan Ventura, and Mary Lou Maher, 57–64. Association for Computational Creativity, 2019.

Veatch, Thomas C. “A Theory of Humor.” *Humor: International Journal of Humor Research* 11, no. 2 (1998): 161–215.

Vechtomova, Olga. “UWaterloo at SemEval-2017 Task 7: Locating the Pun Using Syntactic Characteristics and Corpus-Based Metrics.” In *Proceedings of the Eleventh SemEval International Workshop on Semantic Evaluation*, edited by Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens, 421–425. Stroudsburg, PA: Association for Computational Linguistics, 2017.

von Ahn, Luis, and Laura Dabbish. “Designing Games with a Purpose.” *Communications of the ACM* 51, no. 8 (2008): 58–67.

Wardrip-Fruin, Noah. *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. Cambridge, MA: MIT Press, 2012.

Wardrip-Fruin, Noah. “The Story of Meehan’s *Tale-Spin*.” Accessed August 31, 2020. <https://grandtextauto.soe.ucsc.edu/2006/09/13/the-story-of-meehans-tale-spin/>.

Weitz, Chris, and Tony Gilroy. “Rogue One.” [Script.] Accessed August 31, 2020. https://transcripts.fandom.com/wiki/Rogue_One:_A_Star_Wars_Story.

Weizenbaum, Joseph. “ELIZA—A Computer Program for the Study of Natural Language Communication between Man and Machine.” *Communications of the ACM* 9, no. 1 (1966): 36–45.

Weizenbaum, Joseph. *Computer Power and Human Reason: From Judgment to Calculation*. New York: Freeman, 1976.

Welcome.AI. “Comedy Bot—IBM Watson Chatbot Competition.” Accessed November 26, 2020. <https://www.welcome.ai/projects/comedy-bot-ibm-watson-chatbot-competition>.

West, Robert, and Eric Horvitz. “Reverse-Engineering Satire, or ‘Paper on Computational Humor Accepted Despite Making Serious Advances.’” In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, edited by Pascal Van Hentenryck and Zhi-Hua Zhou, 7265–7277. Palo Alto, CA: AAAI Press, 2019.

Whissell, Cynthia. “The Dictionary of Affect in Language.” In *Emotion: Theory and Research*, edited by Robert Plutchik and Henry Kellerman, 113–131. San Diego, CA: Harcourt Brace, 1989.

Wilson, Deirdre. “The Pragmatics of Verbal Irony: Echo or Pretense?” *Lingua* 116 (2006): 1722–1743.

Wilson, Deirdre, and Dan Sperber. *Meaning and Relevance*. Cambridge: Cambridge University Press, 2012.

Woodman, Geoffrey F. “A Brief Introduction to the Use of Event-Related Potentials (ERPs) in Studies of Perception and Attention.” *Attention, Perception and Psychophysics* 72, no. 8 (2010): 2031–2046.

Yahr, Emily. “Roseanne Barr on Her Valerie Jarrett Tweet: ‘I Was So Sad That People Thought It Was Racist.’” *Washington Post*, July 27, 2018. <https://www.washingtonpost.com/news/reliable>

-source/wp/2018/07/26/roseanne-barr-on-her-valerie-jarrett-tweet-i-was-so-sad-that-people-thought-it-was-racist/.

Youtube. “Double Vision.” Accessed August 31, 2020. <https://www.youtube.com/channel/UChA8UAl8GChxNqXTqrKzTow/videos>.

Youtube. “Star Trek Generations.” [Clip.] Accessed August 31, 2020. [Clip]https://www.youtube.com/watch?v=vffF4Jq478_s.

Zillmann, Dolf. “Disparagement Humor.” In *Handbook of Humor Research*, edited by Paul E. McGhee and Jeffery H. Goldstein, 85–107. Berlin: Springer-Verlag, 1983.

Ziv, Ziversus “Humor as a Social Corrective.” In *Writing and Reading across the Curriculum*, 3rd ed., edited by Laurence Behrens and Leonard J. Rosen. Glenview, IL: Scott Foresman, 1988.

Index

- ABBA structure, x
Abbott, Lou, 2, 77
Abbott & Costello, 2, 77
Absurdity, 27–28, 39, 44, 170, 198, 221
Acquisition of words, age of, 169
Adams, Reginald, 39
Adaptation, evolutionary, 39
Adventure games, 83
Aesthetics, 31, 43, 133, 231
Affective computing, 248. *See also* Sentiment
Aggression, 11–12, 15, 50
Aha! moment, 33, 43, 106, 239
AI-Complete, 8
Air quotes, 216
Alexa, Amazon service, 87, 88, 93–95, 98, 100–101, 248
Alliteration, 3, 185
Alpha 60 (fictional AI), 131
Alphaville (film), 131
Altshuller, Genrich, 117–118
Amadeus (film), 201–202
Amazon, 59, 62, 87, 120
 Alexa service, 87, 88, 93–95, 98, 100–101, 248
 book recommendations, 59, 62
 Echo device, 87, 99
Mechanical Turk service, 164, 167, 197
Ambiguity, xiv, 112, 166–167, 184–187, 201–204, 231
 lexical, 184, 231
 structural, 186
AMT. *See* Amazon: Mechanical Turk service
Analogy, 36, 58, 113–114, 133, 190–192, 246
 false analogies, 190
 faulty analogies, 192
AnalyzeWords.com, 214
Angles, in jokes, ix, 20, 107, 109, 114, 120
Anomalisa (film), 216
Anthropomorphism, 87, 94, , 99, 103
Anti-Semitism, 234, 238
Antibodies, 227
Antimetabole, x
Anti-weight (TRIZ), 118, 120–122
Appropriate incongruity. *See* Incongruity:
 appropriate
Archimedes, 33
Aristotle, 26–27, 38, 55
ARPABET phonetic alphabet, 155
ARPANET, 80
Artificial intelligence, xi, 1, 133
ASCII, 173
Asimov, Isaac, 106
Association, conceptual, 114, 133, 137
Asymmetry (TRIZ), 119
Attardo, Salvatore, 36, 39
Attention mechanisms, 148

- B-9 (fictional AI), 2, 37, 78
Back-off strategies, 81, 153–154
Balanced data splits, 188
Barr, Roseanne, 237–238
Bayes, Rev. Thomas, 17
Bayesian inference, 17, 18
Beam search, 86
Beattie, James, 28
Beauty, 43, 54
Benign behavior, 2, 10, 13, 38–39, 50–51, 54, 205, 225, 242, 244
Benign violation, the theory of, 38–39, 205, 242
Bergen, Edgar, 78, 80
Bergson, Henri, 29, 30, 34, 36, 38, 51, 93, 99, 102, 133
Beyond the Fence (stage show), 247
Bhatnagar, Ranjit, 54
Bialystock, Max, 115
Big data, 240
Binsted, Kim, 167
Bisociation, 33, 36, 49, 106, 113
Blackboard architectures, 88, 95
Blacklist, of words, 171, 228–234
Blade Runner (film), 204
Blessing in Disguise (TRIZ), 123
Blue list, of words, 232
BNC. *See* British National Corpus
Boeing 737, 119
BookCorpus (resource), 164
Bootstrapping, of AI systems, 208
Botnik Studios, 200
@botOnBotAction, 230
Bots, 20, 44, 200. *See also* Chatbots
Brexit, 247
British National Corpus, 181, 183
Brooks, Mel, 31, 116
Buckingham, George, 69, 75
Burlesque theater, 77
Burroughs, William S., 52, 106, 115, 125, 243
BuzzFeed, 112
C-3PO (fictional AI), 78, 87
Caine, Michael, 184
Call-response formula, 76
Captions, of cartoons, 194–199
Canned responses, 234
Carlin, George, 107, 125, 225, 228–229
Cartoons, 194–199
Catastrophe theory, xv, 135–146, 193–194
Catchphrases, 3–5, 8
Category errors, 30–31
Catharsis, 4. *See also* Relief
Causal graphs, 89
Center embedding, 69–70
Cerf, Vint, 80
Chabon, Michael, 201
Chandler, Raymond, 111, 219
Chaplin, Charlie, 30
Charniak, Eugene, xiii
Chatbots, 19, 78, 222, 234, 239
CheapBotsDoneQuick.com, 69, 73, 75
Cheney, Tom, 195
Chess, 15, 80, 129
Child’s humor, 47
Churchill, Winston, 222
Clark, Herbert H., 206
Clichés, 114–115, 246
Clippit, 17
Clippy (software), 17
Closed-world reasoning, 50, 75
Clustering, of words, 196
CMU Pronouncing Dictionary, 155, 160
Cognitive modeling, 22
Coherence, of narratives, 92, 223
Colby, Kenneth, 79–80
Color Changes (TRIZ), 123
Colossus: The Forbin Experiment (film), 14
Colton, Simon, 247
Common sense, 8, 26, 38, 60, 131, 138, 145, 183, 220
Compartmentalization, 127, 130
Composite Materials (TRIZ), 120
Compressibility, 42, 43

- Compression, of data, 41
Compton, Kate, 68
Conceptual blending, 31, 33, 62, 114, 127, 246
Concision, 111
Concreteness, of words, 169
Conditional probability, 153–154
Conflict, mental, 11–15, 28, 34, 48, 78–80, 91, 128–132, 148, 166, 244
Connectionist approaches, 211. *See also*
 Neural networks
Conspiracy theories, 35, 234, 237
Consumer perspective, the, 193
Context, 5, 16–22, 39–40, 48, 51–52, 61, 109–110, 114, 123, 140–142, 145, 149, 151–162, 178, 182–183, 194, 205–206, 212–214, 227–228, 232–233, 237–244
Controlled laughter, 40
Conventional wisdom, 10
Convergence (game), 176
Convolution, in neural networks, 211–212
Cook, Mike, 200
Cooperative principle, the, 205
Corden, James, 120, 123–124
Cosine similarity, 185
Cost models, 15, 155, 189
Costello, Bud, 2, 77
Coulson, Seana, 32
Counterfactuals, 32, 187
COVID-19, 18
Crayola, 124
Crosby, Bing, 93
Crowdsourcing, 164, 218
Crowdflower. *See* Figure Eight
Cusp, geometric, 135, 143, 193
Customer service agents, 19, 241
Cut-up method, the, 52, 69, 106, 227, 243
CYC knowledge base, 245
Cynics (philosophy), 25

Dalí, Salvador, 105–106
Data, Commander (fictional AI), 9, 12
Data structures, 35
Day jobs (for AI), 23, 243, 248
dbpedia.org, 57, 59
Dean, Greg, 114, 135
Decoders, 176
 decoding process, 175–177, 181
Deep learning, 11
Deep networks, 11–12, 211, 240
@deepDrumpf, 64
Default interpretation, 216
Dennett, Dan, 39
Detachment, 29, 80, 121, 127
Deterministic processes, 177
Dialogue, 19–20, 78–79, 84, 98, 102, 222–224, 239
Diogenes, 25, 28, 29, 30
Discontinuity, geometric, 135
Disneyland, 32, 36
Disproportionality, 28
Distinctiveness, 166
Domain-model/domain-level, 95
Doogan, Sam, 155
Double acts, 76, 93
Double take, 83, 138, 189
Double vision, 113, 122
Downshifting, of sentiment, 219
Dramatic irony, 204
Dreams, interpretation of, 66
Dropout, in neural networks, 212
Dual process theory, 136
Dubbin, Rob, 228
Duchenne laughter, 40
Duluth (system), 158, 161
Dynamics (TRIZ), 118, 122
Dystopian scenarios, 130, 248

Echo (Amazon device), 87
Echo, of form, 63, 150, 153, 204–206, 208
Echoic mention, 208, 216–217
Edinburgh Festival Fringe, 34
Edison, Thomas, 106
Edit distance, 154–155, 190

- Educated insolence, 26, 48, 55, 130
- EEG. *See* Electroencephalogram
- Ego, 3, 27, 80
- 8chan (forum), 236
- Electroencephalogram, 138
- Eliza, 78–84, 98, 222, 239
- Eliza effect, 84–85, 93, 95
- Embodiment, 41, 87, 93–94, 101–103
- in physical performance, 99–104
- Emergence, in complex systems, 13, 16
- Encoders, 176
- encoding process, 175–177, 211
- @enjoyTheMovie, 51
- Entropy, 166
- ERP, 139–140
- Euclidean distance, 210
- Eureka moment, 33, 106. *See also* Aha! moment
- Event-related potentials. *See* ERP
- Evolution, of humor, 39, 41
- Expectedness, 144–146, 193
- Explainable AI, 240
- F₁ score, 158–159, 215
- Facebook, 234
- Fallon, Jimmy, 120
- False analogies, 190
- False laughter, 40
- False negatives, 228
- False positives, 228
- Fauconnier, Gilles, 31, 127
- Fear, 17, 31, 59, 79, 142, 248
- Feature engineering, 182
- Feed lines, 183. *See also* Punch lines
- Figaro (system), 223
- Figure Eight (service), 218
- Figure-ground reversal, 36, 121
- Fitzgerald, F. Scott, 127–128
- Flouting, of maxims, 205
- Formulaic jokes, 48, 75–76
- 4chan (forum), 236
- Fracking, of sarcastic intent, 212
- Frames, 5, 33, 37, 42, 48–49, 57
- frame shifting, 32, 34
- framing, of a joke, 70–72, 111
- Frankenstein, Victor, 31
- Frasier* (TV show), 30
- French Toast* (game), 178–179
- Freud, Sigmund, 30–33
- Fun, 9–10, 13, 43, 48
- Funny man (in comedy), 26, 77–78, 80, 221
- Futurama* (TV show), 247
- Games
- adventure games, 83
- games with a purpose, 189–191
- gamification, 21, 106
- serious games, 189–191
- Gender, 56, 142, 235, 239
- General theory of verbal humor, 36–37
- Generate-and-test approaches, 108
- Generative pretrained transformer. *See* GPT architecture
- Generator-generators, 73
- George & Gracie, 2
- Gerrig, Richard J., 206
- Getty, J. P., 52
- Gestures, 94, 99, 103
- Ghosh, Aniruddha, 212–215
- Giora, Rachel, 141, 216
- GloVe vectors, 180–181, 211
- Godard, Jean-Luc, 131
- Gödel, Kurt, 132, 174
- Gödel number, 174–175
- Gödel statement, 132–133
- Gödel's proof, 133
- Goodman, Noah, 165
- Good sense of humor, 1, 9, 12, 249
- Goodwin, Ross, 244
- Google, 52, 180, 220–221, 227
- n-grams, 150, 158, 162, 185, 223–224
- GPT architecture, 86, 90, 147–148
- Graph structures, 36–37, 49, 89, 107–108, 211
- in causal models, 84, 89–90, 92, 224

- Gray list, of words, 229–231, 239
 Grice, H. P., 205
 Gricean pragmatics, 205, 208
 Groaner puns, 149
 Ground truth, 190, 197–199, 209, 213
 Grounding, of words, 196–197
 GSOH. *See* Good sense of humor
 GTVH. *See* General theory of verbal humor
 Gurevych, Iryna, 155, 160
 GWAPs. *See* Games with a purpose
 Gysin, Brion, 52, 106, 115, 125
- HAL 9000 (fictional AI), 11, 17, 128–129
 Handles, of jokes, 110, 114, 120
 Hao, Yanfen, 220
 Hate speech, 228
Have I Got News for You (TV show), ix
 He, He, 161
 Headlines, 52–56, 188–193, 226–227
 Hearst, Marti, 220
 Heinlein, Robert, 9, 12
 Helm, Nick, 157
 Hempelmann, Christian, 155
 Heterographic puns, 155
 Hobbes, Thomas, 27, 128
 Hofstadter, Douglas, 84, 133
 Holocaust, 232, 235
 Holographic memory, 129
 Home-care companions, 18, 241
 Homographic puns, 155, 158
 Homophonic puns, 167
 Hope, Bob, 93
 Hope & Crosby, 93
 Horror (genre), 31, 33, 142
 Horvitz, Robert, 189–191, 195–199
 Humor Research Lab (HuRL), 38
 Hurley, Matthew, 39
 Hutcheson, Francis, 28
 Hypnagogic state, 105
- Iambic pentameter, 54
 IBM, 21
- Idioms, 10, 246
 Idiom Savant (system), 156, 158–159, 161–162
 Imageability, of words, 169, 187
 Image schemas, 103
 Imagination, moral, 226
 Immune system, 234–236
 Imprecision, 57
 In-groups, 17
 Inappropriateness, 39, 52, 64, 204, 248
 Incompleteness, 132
 Incongruity, 28, 33, 37, 44, 48, 91, 106, 118, 127–128, 133–134, 138, 140
 appropriate, 39, 52–53, 59–60, 80, 90–91, 183, 226
 bots, 52
 resolution, 39, 242
 Inconsistency, 127, 132
 Innuendo, 153, 227
 Insight, 41, 119
 Insincerity, 202–203, 207
 Inspiration, ix, 108
 Intent model, of Alexa, 94
 Interactive fiction, 83
 Interpretation, of jokes, 158
Interstellar (film), 2, 14, 134
 Intuition, 137, 240
 Irony, 19, 39, 62–63, 71, 120, 172, 187–188, 200–201, 232
 #irony hashtag, 204, 215, 217–219
 Izzard, Eddie, 6
- Jaech, Aaron, 155
 JAPE, 168–172
 JAPE-1, 169
 JAPE-2, 169–170
 Jarrett, Valerie, 238
 JavaScript, 68
 Jenner, Caitlin, 227, 237
 Job (Bible book), 236
 Joke Appreciation & Publication Engine.
 See JAPE

- Jokesplaining, xi
Jolie, Angelina, 227
Joy, internal, 43
JSON. *See* JavaScript
Jump scares, 33
Jurassic Park (film), 32, 36
- K-2SO (fictional AI), 9, 12
k-nearest neighbors, 210
k-NN (algorithm). *See* k-nearest neighbors
Kafka, Franz, 182
Kahneman, Daniel, 137
Kant, Immanuel, 27, 43, 134
Kao, Justine, 165, 167
Kaufman, Charlie, 216
Kazemi, Darius, 52, 226–228, 231
Keats heuristic, 54–55
Kingston, Miles, 82
Kirk, Captain, 128, 149
Knight, Heather, 104
Knock-knock jokes, 47–48, 72–73
Knowledge
 ontological, 37, 47, 53, 56–57, 65–70, 82, 90, 109, 245
 procedural, 37, 47, 56, 82
 shallow, 56, 59, 84, 222–223, 246
Koestler, Arthur, 33, 36, 127
Koncel-Kedziorski, Rik, 155
Kubrick, Stanley , ix, 128–129
Kullback-Leibler score, 167
- Language models, 20, 22, 106, 144, 150, 152, 156, 234, 242, 246
 neural, 86, 90, 147–148
 smoothing in, 152, 154, 193
Latent meaning, 172, 180, 181
Latent semantic analysis, 180–181, 183–184, 211
Latent similarity, 181
Lateral thinking, 246
Laughter, 1–3, 27–30, 38–43, 248
Laurel & Hardy, 2, 77, 93
- Le Guin, Ursula K., 142
Le Rire (book), 29
Learning
 deep learning, 11–12, 64, 211, 240
 machine learning, 16, 19, 182, 207–210, 222, 240
 one-shot, 4, 5
 supervised learning, 182, 207, 211
LEGO, 6, 7, 58, 159
Leibniz, W. G., 175
 Leibniz number, 175
Leith, Sam, 131
Lenat, Douglas, 245
Leno, Jay, 110
Leviathan (book), 27
Levy, Roger, 165
Lewis & Martin, 2, 93
Lewis, Jerry, 2, 77, 93
Lexical ambiguity, 184
Liang, Percy, 161
Lightyear, Buzz, xiv
Linguistic inquiry and word count, 214
Literal meaning, 20
Little Britain (TV show), 3
LIWC (resource). *See* Linguistic inquiry and word count
- LMs. *See* Logical mechanisms
Logical mechanisms, 36, 121, 190
Long short-term memory, 212
Lost in Space (TV show), 2
LSA. *See* Latent semantic analysis
LSTM. *See* Long short-term memory
- Magic, xi, 39–40, 55, 62, 78, 85
 sympathetic magic, 39–40, 55
Mankoff, Bob, 194–200
Manurung, Ruli, 171
Marsh, Will, 34
Martin, Dean, 2, 77, 93
Matrix (mental), 33, 36, 127
McCarthy, Charlie, 78, 80
McGlone, Matthew, 54

- McGraw, Peter, 38
- Mechanics Substitution (TRIZ), 121
- Meehan, James, 83
- Meldrew, Victor , 3, 7
- Memory, 10, 59, 129, 130, 146, 212–213
holographic, 129–130
long short-term, 212
- Mental states, 4, 8, 106
- @mentions, of Twitter handles, 230
- Merging (TRIZ), 120, 123
- Meta-model/meta-level, 82, 94, 98, 102
- @metaphorMagnet, 217–218
- Metaphors, 56–57, 102, 110, 201, 245–246
- Meyers, Seth, 121, 123
- Microsoft, 17, 195, 234–236
Microsoft Word, 17
Microsoft Research, 195
- Mihalcea, Rada, 183, 185, 188
- Mike (fictional AI), 9, 12, 137
- Mildner, Thomas, 93–94
- Miller, George, 157
- Miller, Tristan, 155, 160
- Minsky, Marvin, 38
- Minstrel (system), 83
- Misogyny, 234
- Misspun tales, 84
- Modularity, of humor, 9, 12, 37
- Monroe, Marilyn, 110
- Montfort, Nick, 176
- Morality, 8, 13, 30, 37, 226, 239
- Morreall, John, 142
- Motivation, 44, 203
- Mozart, Wolfgang Amadeus, 201–202,
206
- Multitasking, 95
- N400, 139–140, 143
- n-grams, 150, 187
- N+V theory, 38, 39
- N-word, the, 235
- Nagel, Ernest, 133
- NAO robot, 87–88, 91–104
- Negation
explicit, 208
implicit, 213, 219
- Negative exemplars, 186–188
- Networks
artificial neural networks, 11, 22, 64,
146–147, 211, 214–215, 240, 245
network effects (in AI), 19
semantic networks, 35, 108–111, 168
- Neural networks, 11, 22, 64, 146–147, 211,
214–215, 240, 245
- New Yorker, The*, 194
- Newman, James, 133
- Nietzsche, Friedrich, 54–55
- NOC list, 49, 58, 61, 65, 90–91
- Non-default interpretation, 216
- Non sequiturs, 245
- Nonliteral language, 207
- Non-self-discrimination, 236
- Nonsense, 10, 44, 60, 132, 221
- Nonseriousness, 142
- Norms, 38, 57
- #NOT hashtag, 208, 219
- Novelty, 42–43, 120, 144, 148
- Null context, 218
- O’Brien, Conan, 111–112, 121
- O’Hara, Kenton P., 235, 239
- @oldSkoolFunBot, 63
- One Foot in the Grave* (show), 3
- Onion, The* (online publication), 188–192,
196
- Online reviews, 207, 209
- Ontological semantic theory of humor, 37,
90, 109
- Ontology, 25, 56, 90
- Open AI, 86, 148
- Open-world reasoning, 50, 75
- Opposition, semantic, 35, 78, 110, 191, 192,
204–205, 208, 218–219
- Optimal innovation, 141, 149, 153–154,
156, 164, 188, 190, 242–243, 246, 248

- Oring, Elliott, 39, 53
- Orthodoxy, 1, 27
- Orwell, George, 39
- Ostendorf, Mari, 155
- OSTH. *See* Ontological semantic theory of humor
- Other Way Around (TRIZ), 123
- Otherness, 44
- Out-groups, 17
- Oxymorons, 39
- Pain, Helen, 171
- #PaintMySoul hashtag, 230
- Pantomime, 94, 99, 216
- Parameter Changes (TRIZ), 118
- Paranoia, 83, 240
- paranoid schizophrenia, 79
- Parody, 64
- Parrish, Alison, 237–238
- Parry (system), 79–83, 98, 222, 239–240
- Parts-of-speech tags, 186, 197–199
- Pascal, Blaise, 28
- Paulos, John Allen, 134
- Paxton, Bill, 4
- Peng, Nanyung, 161
- @pentametron, 54–55
- Perfect puns, 155, 167, 186
- Perplexity, 185, 197–199
- Personality types, 136, 213
- Personification, 84
- Philips, Emo, 32
- Phonetic similarity, 149, 153
- Physics envy, 183, 185, 187
- Pinker, Steven, 203
- Planet of the Apes* (1968 film), 204, 238
- Plato, 25, 28, 29, 37
- Pleasure, 38–39, 50
- pleasure principle, the, 38
- PMI. *See* Pointwise mutual information
- Poetic justice, 204
- Pointedness, 187
- Pointwise mutual information, 159, 163
- Poker, 203
- Polanski, Roman, 31
- Polythetic concept, irony as, 206
- Pornography, 39
- Portmanteaus, 19
- POS tags. *See* Parts-of-speech tags
- Precision, measure of, 158, 160, 212, 215
- Predictability, 1, 41
- Prejudice, 234, 239
- Pretense, 15, 63, 204, 206, 219
- view of irony, 206
- Pretraining, 180
- Priming effects, 137, 141–142, 156
- Probability distribution, 152, 166, 194
- Producer perspective, 193
- Producers, The* (film), 116
- Profanity, 227
- Project Gutenberg, 178
- Propositional attitude, 200–203, 246
- Propositional content, 82, 200–203, 217, 246
- Psycholinguistic data, 169, 187
- Pullman, Stephen, 183
- Punch lines, 8, 32, 43, 48, 111–114, 123–124, 137, 163, 191
- Puns, 21, 47, 114, 149, 155–158
- groaners, 149
- heterographic, 155, 160
- homographic, 155, 158–159
- homophonic, 167
- perfect, 155, 167, 186
- Pygmalion* (play), 79
- Python (programming language), 228
- Quality, maxim of, 205
- R2-D2 (fictional AI), 78, 87
- Race conditions, 138
- Racism, 234, 239–240
- Randomness, 52, 63, 69–72, 106, 158, 186, 202
- random baselines, 184–185
- random walks, 89–92

- Raskin, Victor, 33–37, 127, 241
Readability, of text, 198–199
@readMeLikeABot, 58, 232–233, 237
Recall, measure of, 158, 160, 212, 215
Recommender systems, 62
Recurrent neural networks, 64, 146, 245
Red list, of words, 232
Reddit, 207
Regulator, inner, 226–227
Regulator, outer, 226
Release, psychological, 31. *See also* Relief, in humor
Relevance, 17, 39, 112, 157, 205
relevant inappropriateness, 39
Relief, in humor, 28, 38, 44, 93
Resolution. *See* Incongruity: resolution
Restraint, prior, 226
Reversal, in jokes, 36, 121
Reward
 aesthetic, 43
 functions of, 21, 38–39, 189–190
Reyes, Antonio, 185, 188
Rhyme, 54–56, 185
Riddles, 131, 149, 156, 164, 168–170
Rieg, Stefan, 92
Rigidity, 1, 27, 51, 93, 99, 102–103
Riloff, Ellen, 208, 212
Ritchie, Graeme, 168, 171
Road Runner (cartoon character), 2
Robot & Frank (film), 18, 129
Rogue One: A Star Wars Story (film), 9
Rollerball (film), 130
Rosso, Paolo, 185
Rube Goldberg devices, 212
Rudy (system), 239–240
- Salience, 48, 117, 121–122, 141, 158
Salieri, Antonio, 201–202, 206
Sarcasm, 15, 18–19, 202–203, 207–215, 220
 #sarcasm hashtag, 207–208, 215
SASI (system), 209–210
Satire, xi, 31, 63, 66, 116, 188–193, 206
Scare quotes, 216–219
Scéalextric (system), 89–95, 99
Schadenfreude, 102, 248
Schematic reasoning, 94, 122–123, 168–171, 243
 image schemas, 103
Schlesinger, Ari, 235, 239
Schmidhuber, Jürgen, 41, 144
Scholastic Aptitude Test, 191
Schopenhauer, Arthur, 28
Scooby Doo (robot), 16
Scripts, 5, 10, 33–34, 42, 48–49, 57, 82
 script opposition, 35, 116, 120, 199, 218–219
 script triggers, 7–8, 17
 script-switching, 34
Scunthorpe problem, 231
Seinfeld (TV show), 30, 238, 243, 245
Seinfeld, Jerry, 107, 109, 125
Self-awareness, 12
Self-deprecating humor, 233
Self-discrimination, 236
Self-reference, 175
Semantic grammars, 67, 242
Semantic networks, 108, 168
Semantic script theory of humor, 33, 34, 127
SemEval, 155, 160, 163, 164
Sentiment, 5, 20, 187, 198–199, 217, 230
 affective computing, 248
 analysis of, 214, 230
Serendipity, 44, 54–56, 63
Serious games, 189–191
Serling, Rod, 238
Setup, for a joke, 8, 111–115, 122–124, 140–141, 144, 153, 183, 214
Sexism, 227, 240
Shaftesbury, third Earl of, 28, 29, 93
Shahaf, Dafna, 195–199
Shakespeare, William, 54
Shame, 79, 207
Sharp, Oscar, 245
Shaw, George Bernard, 79

- Shelley, Mary, 31
Shock value, 37
Similarity, measures of
 cosine similarity, 185
 edit distance, 154–155, 190
 latent similarity, 181
 phonetic similarity, 149, 153
 word similarity, 163
Similes, 57, 65, 217, 220–221, 232–233
Simplicity, subjective momentary, 42
Sincerity, 202, 216
Singular value decomposition, 181
Situational irony, 203
SJWs. *See* Social justice warriors
Skelton, Charlie, ix–x
Skip-grams, 163, 187
Smith, Dr. (*Swiss Family Robinson* character), 2, 78
Social justice warriors, 236
Softbank Robotics, 87
Space, abstract possibility, 10, 83, 88, 107, 125, 134–136
Spaces, mental, 32, 33, 127
Spatial metaphors, 102
Sperber, Dan, 205, 216
Spock, Mr. (*Star Trek* character), 128
Spoilers, movie, 51
SpongeBob SquarePants, 78
SSTH. *See* Semantic script theory of humor
Stance-taking (in AI), 234
STANDUP (system), 171
Star Trek, 9, 128, 149
 Star Trek: Generations, 9
 Star Trek: The Next Generation, 9
Star Wars (film), 6, 9–10, 87, 94
Starbucks, 108–114, 169
Statistical processing, xii-xiv, 19–20, 37, 106, 240
 AI, statistical winter of, 37, 241
@stealthmountain, 50
Stereotypes, 3, 137, 221
Stories, 83–95
Straight man (comedy role), 26, 77, 221
Strapparava, Carlo, 183
Strong Oxidants (TRIZ), 122
Structural ambiguity, 186
Stupidity, 38
Subjectivity, 10, 42, 58, 239
Subversion, 26, 61, 71, 170
Sun, Jonny, 54
Sunspring (film), 245
Superiority complex, 15, 50, 128, 131
Superiority humor, 27, 30–32, 44, 50–51, 64–65, 128, 230, 244
Supernormal stimuli, 39
Supervised learning, 182, 207, 210–213
Surprisal (measure), 162, 164
Surprise, 144, 162, 164
Suspension of disbelief, 55, 206
Symbolic processing, 11–12, 26, 37, 89–95, 168, 172, 240–242
 subsymbolic processing, 12, 176
Sympathetic magic, 39–40, 55
Synset, in wordnet, 157, 169
Systems 1 and 2 (Kahneman), 137, 148, 240
Taboos, 29, 225, 228, 235
Taking Out (TRIZ), 122
Tale-Spin (system), 83
 Tale-Spin effect, 84–85, 87, 91, 93
TARS (fictional AI), 2, 14–16, 78, 133
Tay (system), 234–235, 238
 @TayAndYou. *See* Tay
Taylor, Alex S., 235, 239
Taylor, Archer, 220
Taylor, Nathan, 247
Teasing, 202
Templates, for jokes, 47–50, 63, 72–75, 168, 187, 222
Temporal compression, 187
Temporal imbalance, 187
Terminator, The (film), 4
TF.IDF (measure), 181
Theory of mind, 7, 13, 15, 142, 244

- Till, Benjamin, 247
@tinycarebot, 54
TIPS. *See* TRIZ
Tofighbakhsh, Jessica, 54
TOM. *See* Theory of mind
Toplyn, Joe, 108–112, 114–115, 120, 124, 163
Toy Story (film), xiv
Toy systems in AI, 169
Tracery, 67–70, 73–75
Training (of AI systems), 11, 207, 210–213
 training data, 20, 64, 147–148, 198,
 209–213, 240
Transgression, 38, 242
Triggers, of behavior, 7–8, 17, 51, 59, 69,
 82–84, 124
 script triggers, 7–8, 17
TRIZ, 21, 117, 121–122, 125, 247
Tropes, 31, 59–60, 188, 228, 238
Trump, Donald, 64, 90
 @TrumpScuttleBot, 64
Truth, 32, 54–55, 65, 102, 105, 115,
 131–138, 183
 ground truth, 190, 197–199, 209, 213
Tsur, Oren, 209, 212
Tudyk, Alan, 10
Turner, Mark, 31, 127
Turner, Scott R., 83
Twitter, 44, 204. *See also* Twitterbots
Twitterbots, 44–45, 47, 50, 69, 200, 207,
 213–214, 217, 226, 230
Two-stage model of interpretation, 137
2001: A Space Odyssey (film), 11, 14, 128
Two-way words, 229
@twoheadlines, 52–53, 56, 226
- Unbalanced data splits, 188
Uncanny. *See* *Unheimlich*
Understatement, ironic, 222
Unfun.me, 189
Unheimlich, 30, 198
Unicode, 173
Universality (TRIZ), 120, 123
UWaterloo (system), 159
- Valitutti, Alessandro, 218
Vaudeville theater, 77
Veatch, Thomas, 38–39
Vector space, 175–177, 183, 199,
 210–211
Ventriloquism, 78
Verbal irony, 203, 232
Vidal, Gore, 217
Violations, 37
Virtual tutors, 21, 241
- Walkie-Talkie (system), 87, 95
Wardrip-Fruin, Noah, 84
Warhol, Andy, 217
Warren, Caleb, 38
Watson AI (system), 21
Weaponization, of AI systems, 236
Web-scraping, 57
Weizenbaum, Joseph, 78
West, Bob, 189–190
West, Kanye, 48–49, 189
What-If Machine (system), 247
WHIM project. *See* What-If Machine
White list, of words, 231
Wicke, Philipp, 93–94
Wikipedia, 57
Wilder, Gene, 31
Wile E. Coyote, 2
Willis, Bruce, 227
Wilson, Deirdre, 205, 216
Wine speak, 111
Word sense disambiguation (WSD),
 157–159, 161
Word similarity, 163
WordFilter, 228–229, 231–233
WordNet, 157–158, 160, 169
Wordplay, 19, 149, 157
Word2Vec, 159, 180–181, 211
WSD. *See* Word sense disambiguation
- #yeahrighat hashtag, 208, 213
- Zero (fictional AI), 130

