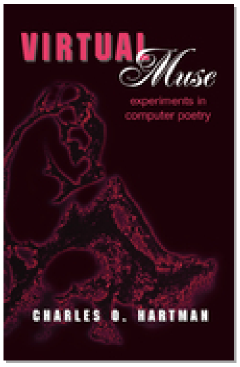
Virtual Muse

Charles O. Hartman



Published by Wesleyan University Press

>>>> / 5 /

TRAVESTY

In November 1984, Hugh Kenner, the great Modernist literary critic, and his colleague at Johns Hopkins, the computer scientist Joseph O'Rourke, published an article in *Byte* magazine called "A Travesty Generator for Micros."

Neither literary scholars nor computer science professors publish often in microcomputer magazines. Kenner and O'Rourke were both playing hookey, from different schools. (It's worth noting, though, that Kenner had already written on Buckminster Fuller's mathematics and on the relevance of group theory to poetry.) Naturally enough, the article's claims on its readers' interest are divided.

On the one hand, the authors are demonstrating the implementation of algorithms (that is, computational procedures) for pattern matching that were first suggested by Brian P. Hayes in the "Computer Recreations" column of *Scientific American* a year earlier. Most of the *Byte* article is devoted to details of the algorithm and its implementation, with discussion of ways to improve efficiency. The article inspired an unusual number of letters to *Byte*'s editor, and most of them offered programming improvements and alternatives.

On the other hand, the purpose of the program itself is to generate "travesty" texts from other texts so as to examine the relation between the original and its transformation and deduce various things about the language of the original. In short, Travesty is a computer stylistics program.

Here's what the program does. A text, such as a passage from a novel, is among other things a set of characters. It consists of so many *e*'s, so many *f*'s, and so on. It's also a set of character pairs (so many *ex*'s, so many *ch*'s, etc.) and of triplets (*the*'s, *wkw*'s, etc.), and so on.

For any same-size group of characters—call the size n—it's possible to make a frequency table for a particular text. From that table, another text can be constructed that shares statistical properties, but only those properties, with the first one. That's what Travesty does. It produces an output text that duplicates the frequencies of n-character groups in the input text. To put the same thing the other way, it thoroughly scrambles its input text but only down to level n.

At n = 1 all you get is a mishmash of letters that more or less obeys the usual frequency distribution of English. (E is the most frequent letter, t is next, and so on). If you set *n* equal to 2, the result is slightly more organized gibberish:

> Dengethe pr: o ls h thee. wicach Ye thur. obbug lesila thicatetonoisthate Thrit O athe are. t is: winsict kerprurise, y m? th o mor sty hetseatheancathensous.

The longest pattern-matching string allowed by the original Travesty is nine characters. At *n* = 9, the output text largely duplicates the input text—except for some odd leaps. Here's the beginning of the tenth chapter of the book of Ecclesiastes:

> Dead flies cause the ointment of the apothecary to send forth a stinking savour: so doth a little folly him that is in reputation for wisdom and honour. A wise man's heart is at his right hand; but a fool's heart at his left. . . .

And here's an n = 9 travesty (using the whole chapter as an input text):

> Dead flies cause the ointment of the ruler: folly is set in great dignity, and the end of his mouth is foolishness: and the end of his talk is mischievous madness. A fool also is full of words: a man cannot tell what shall bite him. Whoso removeth stones shall be endangered thereby. If the iron be blunt, and he do not whet the edge, then must he put to more strength: but wisdom is profitable to direct.

The last sentence in this passage remains perfectly intact. If it sounds strange—well, that's one of the program's effects on our reading. On the other hand, the "ointment of the ruler," though it sounds plausible, is a figment. The program has compounded it from "ointment of the apothecary" and "spirit of the ruler" by way of the repeated nine-character run (including blanks), *t of the* .

Formally, as Kenner and O'Rourke point out, "The connection of the output to the source can be stated exactly: for an order-n scan, every n-character sequence in the output occurs somewhere in the input, and at about the same frequency."

The authors use Travesty to make a number of paradoxical points about language. (We may hear Kenner's voice as the dominant one in this section of the article.) The frequency distributions characteristic of English determine, without intervention from a writer's conscious thought, a startlingly large proportion of what the writer writes. "In fact, the language makes three-quarters of your writing decisions for you."

It may not be surprising, then, that even when *n* equals only 3, the emerging output is, though nonsense, clearly *English* nonsense (again the input is Ecclesiastes):

> Deare thy ings. The thy hedge, afte se the whatter: for so mou, alk iroppen due inch an te in to misdo caught; mants oness! The lisdom the not of an ton, and theast for diggerpenning is shalking! By tall the retheat shat his und, woolips sithe eve len thall bableft. A wisdot sloth, and forength him then rings. For of to ing is man's whildigninch him. Deader: He offen rulefter.

"Misdo" is a real word (hence, "misdeed"). So is "ings," a northern English term for "meadow." Just as Lewis Carroll reinvented "gyre" and "slithy," Travesty has reinvented "ings" and "misdo." Neither of these words, nor the equally valid "sithe," "mants," or "shat," appears in Ecclesiastes. But the English dictionary says they could. And the statistical habits of English designate "woolips" and "diggerpenning" as possible words too, which speakers of English happened never to invent.

Naturally, the grip of statistics grows stronger and stronger as *n* increases. Of all the possible four-letter groups (*zxiq, fmup, qtno*), only a tiny minority are available to the writer of English.

And yet free choice remains. Indeed, free choice is all a writer is aware of while writing. But there's more choice than awareness encompasses. Finally, "The *significant* [my emphasis] statistics derive from the personal habits of James, or Joyce, or Jack London, or J. D. Salinger. Each of these writers, amazingly, had his own way with trigrams, tetragrams, pentagrams, matters to which he surely gave no thought." What interests Kenner and O'Rourke about their program is the emergence of these stylistic signatures: "the unexpected fact that essentially random nonsense can preserve many 'personal' characteristics of a source text."

Before inviting us into this sophisticated examination of stylistics, however, Travesty offers more childish pleasures. One of them is implicit in the program's name: It's the wickedness of exploding revered literary scripture into babble. We can reduce Dr. Johnson to inarticulate imbecility, make Shakespeare talk very thickly through his hat, or exhibit Francis Bacon laying waste his own edifice of logic amid the pratfalls of $n = 9$.

Yet the other side of the same coin is a kind of awe. Here is language creating itself out of nothing, out of mere statistical noise. As we raise n, we can watch sense evolve and meaning stagger up onto its own miraculous feet. We can share the sense of wonder that James Joyce aimed at in the "Oxen of the Sun" chapter of *Ulysses*, where the history of the language from grunts to Parliamentary orations unfolds like a morality play before our ears.

Yet it's not clear where this meaning is coming from. Nothing is created out of nothing; and the principles of nonsense discussed in chapter 2 insist that we keep the perceiving reader in his or her place, responsible along with the text and the author for making sense. The reconsideration of these issues belongs to a later stage of my history, and I'll come back to it in the next chapters.

There are two ways to play with this program. If you keep feeding it texts and adjusting n, you can extend the stylistic discoveries of Kenner and O'Rourke—and liven up a late-night party. Or you can take the hint of the printed source code and start tinkering with the program's innards, extending and modifying. As you learn more intimately how the program works (often by making changes that unexpectedly break it), you're bound to start thinking in new detail about what it does.

Messing around with Travesty, I began to see the progressive rousing of sense out of a mindless sea of letters as resembling the evolution of human beings. It certainly wasn't Darwin's version of evolution. Instead, the body—the mechanical corpuscles and ligaments of language—was evolving into mind. Though Darwinian theory has been saddled with this kind of progressivism, it's a misconception. Biological evolution doesn't *go toward anything*, as Stephen Jay Gould often points out. The story I was telling myself was a myth. Any discursive theory behind it would sound like a lot of Cartesian silliness that I didn't believe in for a moment. This didn't stop me from being excited by the drama I was seeing (or making up). The need for *belief* in a myth has never slowed poets down very much; it's what the myth says about things outside itself that matters.

Just at that time I had written a short poem about a famous chess game. Even after I was sure it was finished, the poem didn't seem adequate. For me, that's become a sure sign that the poem at hand is just one section of a larger work. This is a corollary of the Modernist principle of juxtaposition that we've already glimpsed. Most extended modern poems are built in pieces, which the poet composes into a whole without covering up the seams between adjacent parts. I began to add new sections, in different verse forms. They had different starting points, scattered throughout all the reading I was doing at the time; but they converged, as things written at the same time tend to do, under pressure from the same unresolved concerns. As the poem grew, it didn't settle on a clear central theme, but it revealed a constant set of interweaving obsessions: chess, computers, war, the mathematician Alan Turing. None of this process was very unusual for me, and I can't recall exactly when I decided that some Travesty text should be included. But I know I'd been thinking about the Turing Game.

Alan Turing himself called it "the imitation game." The paper that describes it, "Can a Machine Think?" (published in 1950 in the British journal *Mind*), is one of the important documents of this century, touching on mysteries that intrigue us deeply. Yet some of its details haven't been widely noticed.

The physical setup for the game is simple. Two players, A and B, and a person Turing calls the "interrogator" sit in three separate rooms, communicating by teletype or other impersonal means. The interrogator asks questions and tries to identify A and B. In answering the interrogator's questions, player A tries to imitate player B. Or rather, player A tries to convince the interrogator that he or she or it belongs to the same *category* to which player B insists (truthfully) that she or he or it belongs.

The famous main point of Turing's essay begins when he makes player A (the imitator) a computer, and player B (the one who is imitated) a human being. He uses this situation to formulate the previously vague question "Can a machine think?" Can the computer convince the human interrogator that it too is a human being?

First, though, to explain the game, Turing proposes a different version: Player A is a man who tries to pretend he's a woman; player B is a woman and says so. As Turing's biographer, Andrew Hodges, points out, this was relevant to his own experience of sexual ambiguity in homophobic midcentury England.

Hodges calls this first version of the game

> a red herring, and one of the few passages of the paper that was not expressed with perfect lucidity. The whole point of this game was that a successful imitation of a woman's responses by a man would *not* prove anything. Gender depended on facts which were *not* reducible to sequences of symbols. In contrast, he wished to argue that such an imitation principle did apply to "thinking" or "intelligence."

Hodges's fundamental critique of the essay as a whole is that Turing's isolation of human *intelligence* from the body and senses partly belies the nature of *human* intelligence. In this I believe Hodges is right. But

I'm less convinced by his analysis of the "sexual guessing game." I think he misses the point. Using "gender" in too vague a sense, he assumes that the theorem to be proved or ridiculed is that *men are the same as women.* What I think Turing meant was instead that *sociolinguistic behavior—that is, talk—won't reliably distinguish between a man and a woman.*

This is contrary to what many people assume, consciously or unconsciously. Surely *he* couldn't convince us when it came to clothes. Surely (if the game were reversed—notice that Turing doesn't present it symmetrically), *she'd* give herself away on football. The shrewd interrogator won't just ask how the players feel about the Equal Rights Amendment—ask player A how often "she" drives when "her" husband is in the car.

Turing cuts deeply into our sense of what we know about each other. We've come to understand that much of human reality is linguistic. The names we give things control how we see them. We live in words as fish live in water. But how real is this verbal reality? The game asks: If we can't see or touch or hear a person, how certain can we be about the categories we assign the person to? If our talk lets us define ourselves, it also lets us masquerade. This is one of the fascinations people are discovering on the Internet. They hang out with others from twenty-four time zones, whom they will never see, whose voices they will never hear—people who exist for them only as typed words. The opportunities for confusion and outright fraud are balanced by a certain liberation; racism has a hard time with e-mail.

We often treat socially important categories like gender as absolute. Against this, Turing poses his question in terms of *probability*: "Will the interrogator decide wrongly as often when the game is played like this [between human and machine] as he [*sic*] does when the game is played between a man and a woman?" And his ultimate prediction is couched in similar terms: Within fifty years (from 1950) a computer with a billion bits of memory—not so very large by today's standards, about 120 megabytes—will be able to win the game in 70 percent of five-minute trials. There's an implication that this is the rate of success he'd expect in the gender game, too. (Has anyone actually tried the gender game?)

Other categories besides gender come to mind: race, class, and so on. As soon as "knowing" is put in terms of "imitation," a fundamental shakiness shows up everywhere in our humanly constituted reality. As I thought about Turing's paper, these questions began to infiltrate my poem in various ways.

Turing notes that if the second version of the game is reversed, it puts the human being at a serious disadvantage. It's far easier for the computer to mimic human slowness and errors in arithmetic (as in an imaginary sample dialogue he gives) than for a human to emulate a computer's speedy accuracy. Yet the history of our uneasy relations with machines—at least since the industrial revolution and Frankenstein and the first saboteurs, who threw their wooden sandals (*sabots*) into machines that were replacing their jobs—is full of odd instances of people imitating machines. An interesting example is "the Turk" (along with its own imitators, like Ajeeb or Coney Island). This was a nineteenth-century "chess-playing machine"—it once chastised Catherine the Great for cheating—in the likeness of a man, which in fact concealed a legless war veteran: an imitation within an imitation.

At a certain point, when I felt I was done with the little poems that would be sections of the big poem, I conglomerated all of them into one computer file and used that text as input for a series of eight runs of Travesty, with n = 2, n = 3, and so on through n = 9. I had altered the program so that (instead of producing an amount of output determined arbitrarily by the user) it would keep going until a period or question mark coincided with the end of a line. I ended up with eight sections that looked vaguely like paragraphs.

I've oversimplified the process in several ways. In the first place, I began trying all this before I was really done with "my" parts of the poem. Whenever I altered one of those, I had to alter the big input file made up of them all, and of course that would alter all the output sections when I reran the program.

It's worth stopping to ask why I felt I had to do that. The output was nonsense anyway; why not just leave it alone? Who'd notice the difference? One reason was that any discrepancies might show up visibly in the output, though that wasn't very likely. No one was going to do the backtracking—theoretically possible but extremely laborious—that would be necessary to show that an output section was or wasn't properly derived from the combined input.

A better reason was simply that it was my rule. It was a little like adhering to a pattern of rhyme and meter. Or maybe, since infractions were unlikely to be detected, it was more like the rules of solitaire. Artists, like artisans, try to get things *right*, even when it means carving the backs of the gargoyles. Ultimately, I suppose I was so scrupulous about my procedure precisely because I was departing so far from anything my background had prepared me to think of as "writing poetry." I was defending myself, if only to myself, against charges of self-indulgence and laziness. But if you examine this argument closely, you can see that the rules had changed. I was being faithful to something rather new.

A second minor problem that I've skipped over was deciding exactly what would and wouldn't be included in the input text. Arbitrarily, I simplified some punctuation and ignored lineation. All this was to increase the smoothness, the fluidity, and ultimately the plausibility of the output. For instance, omitting all parentheses meant that I wouldn't end up with unmatched right or left parentheses. These were ploys to boost the appearance of sense, the lure of meaningfulness.

By now I knew the title of the poem: "Monologues of Soul and Body." There have been dialogues between the soul and body throughout English literature; Andrew Marvell wrote a wonderful poem of that name. But in this case the body and soul, rather than conversing, talked somehow past each other. The parts I had written myself were the "soul" parts. In the computer output I saw the body constructing itself out of the material of soul, working step by step back toward articulation and coherence. It's a very Idealist poem, and at the same time very Cartesian, and perhaps monstrous.

So as to give my reader at least a hint about what was going on, I decided to open the poem with two epigraphs, one for each group. The "Epigraph of the Body" was Kenner and O'Rourke's formal statement of the relation between Travesty's input and output. The "Possible Epigraphs of the Soul"—plural because the soul was irreducibly in dialogue with itself—quoted the tall-tale teller Maeterlinck in praise of truth. The soul's domain is irony; the body's is reductive fact.

At this point I was faced with a major decision about structure. I decided wrong the first time. I put all eight Travesty paragraphs, in order, together at the end. The result was unreadable—offensively unreadable. My very first reader saved me by pointing this out with considerable force.

The alternative was simply to go one step further in my role as poetic composer: I scattered the eight "Body" sections among the "Soul" sections at what my ear told me were the right places. The vaguely evolutionary idea I had begun with dictated that the eight "Body" sections, though scattered, should remain in order.

This brings me to the final and most interesting problem that I left out in my simplified description of the poem. I've said that I ran my conglomerated input "Soul" file through Travesty eight times. But in fact I was often disappointed with the program's results. If I ran it twice (with *n* the same both times), I could usually choose one of the results as superior to the other.

Superior how? Sometimes an intriguing combination would crop up (from Ecclesiastes: "Dead for yielding!" "A wise madness!"). Sometimes the program would invent a beautiful word ("avathefomitor," "runkin," "andaneld"). Sometimes a phrase belonging to a later "Soul" section would emerge, provocatively foreshadowed, in an earlier "Body" section. But most often, I chose on the basis that ultimately, and not trivially, governs most poetic decisions: I kept the results that *sounded* best. The fragments came together in pleasing tunes, or attractive rhythms, or evocative echoes of half-apprehended thoughts.

The question was, how many choices should I make? If I could choose the better of two, why not the best of twenty? Of two hundred? The way the program generated random numbers happened to put a limit on the number of candidates (65,536 for each value of n); but multiplied by the eight values of n, it was still a number whose human name is Too Many. In the end, I read nonsense all day for several long days; and when I couldn't read any more, I stuck with the best I'd found.