Virtual Muse

Charles O. Hartman

Published by Wesleyan University Press

For additional information about this book
http://muse.jhu.edu/books/9780819572578

Access Provided by New York University at 03/02/13 4:38PM GMT

> > > > / INTRODUCTION /

Talking about computer poetry is almost like talking about extraterrestrial intelligence: great speculation, no examples.

In the hundred-year history of computers, the most wild-eyed prophets have sometimes dreamed of a utopia in which man and machine (sexism reinforced by alliteration) debate on equal terms about the meaning of life and the relative merits of cells and circuits. Perchance they take the occasion to compose and exchange a few sonnets or maybe odes. Poetry in this distant dream serves as a kind of ultimate touchstone of intelligence. The ability to write poems is the talisman by which we'll know that computers have really *arrived*.

This use of poetry isn't surprising. Poetry has great prestige as the sign of Culture—which doesn't make everybody want to read it. When people are looking around for profound examples of what it is they, uniquely, as people, do, poetry gets conscripted as a time-honored and somewhat weary example. It doesn't usually suit people's purposes to define human uniqueness in terms of economics, prayer, torture, or the invention of the weekend.

Yet when poetry is treated as the hallowed repository of sacred Culture, it's dead. No poet worth reading really thinks of poetry as that. A culture worth belonging to has a present and a future, as well as a past. Poetry is something we *do* with language. Or rather, it's a lot of different kinds of things we do with language. It's a place where we can attend to language, as a stadium is a place to attend to the body. And language *is* something that defines people.

Throughout this book I'll be interested in the complicated boundary between what computers can do with language and what they

can't. Obviously (since the book exists), I believe computers can do something worthwhile in the way of poetry. Which brings me back to the first question: Why hasn't there been any computer poetry?

Well, there has—but not much. There was a computer program in California in 1962 that turned out poems. Its caretaker submitted some short verses to *Horizon* (a big glossy magazine mostly devoted to Culture) under the pen name "Auto-Beatnik." I haven't been able to find any later trace of the program or its author, one R. M. Worthy of General Precision, Inc. A couple of other quite similar experiments surfaced in the following ten or twenty years.

Two decades after Auto-Beatnik came *The Policeman's Beard is Half-Constructed* (1984), a book written by a program called Racter. Racter is short for "raconteur." The program's designers, William Chamberlain and Thomas Etter, were especially interested in getting the machine to tell stories and have conversations. At these terribly difficult tasks, Racter turns out tour de force performances—very impressive, very uneven:

> Bill sings to Sarah. Sarah sings to Bill. Perhaps they will do other dangerous things together. They may eat lamb or stroke each other. They may chant of their difficulties and their happiness. They have love but they also have typewriters. That is interesting.

A. K. Dewdney, the human reviewer of Racter's book for *Scientific American*, quoted this example to show, first, how "marvelously funny and even thought-provoking" Racter's productions could be, and then how nonsensical: "The allowances I have been making for Racter all along are stretched to the breaking point when Racter mentions that besides their love they also have typewriters. Invited to share in this extraordinary insight, I tremble on the brink of a completely unknown mental world, one that I would prefer not to enter." As it happened, when I read Dewdney's review, I had already picked this same passage as my favorite example of Racter's potential for profound serendipity. What was going on here?

Well, think about a writer married to another writer. Typewriters,

until computers replaced them not so long ago, were to writers what scalpels are to surgeons. Writers are often cranky people, preoccupied, out of step with everything outside and around them. This makes them difficult to live with—which doesn't make them any better at living with each other. Successful marriages between writers are fairly rare. In the context of such a marriage, Racter's little story would ring loud and true; Bill and Sarah have love, but they also have typewriters.

For one reason or another (human beings lead rather inscrutable lives), Dewdney didn't feel the same resonance that I did. The "completely unknown mental world" that he "would prefer not to enter" wasn't just Racter's, it turns out, but mine as well. In this instance, the computer and I—and not Dewdney—had a meeting of minds.

Or to be more precise, I had the experience of a "meeting of minds," which Dewdney didn't have. There wasn't really a "mind" there for mine to meet. Yet if I were judging a poetry contest and came across Racter's piece, I wouldn't immediately toss it into the Reject pile, and I wouldn't immediately suspect it was a *merely mechanical production*. I might use that phrase to dismiss many other pieces from the book; but for that matter I might reject many of the *human*-authored contest poems in just the same terms (as Cowper did Alexander Pope's). It's important to distinguish "intelligence" like Racter's from real human intelligence, but it's not always easy, not if we go purely on products. If we peek inside the box—look at the program's source code, for example—we may dismiss the whole thing as a contraption. But if we could peek inside the human brain box (which brain science is beginning to do), we might have the same reaction. A neuron has no mind.

The search for understanding between computers and people leads through a denser forest than is usually suggested by the popular literature on "artificial intelligence." People discussing computer intelligence and language and communication have often assumed that we understand human intelligence, language, and communication a lot more straightforwardly than we do.

From one perspective, poetry is a subtle interactive business car-

ried out between a poet and a reader. There are other perspectives, other definitions of poetry, but any theory that leaves out the reader's necessary interpretive genius (though it's often *unconscious* genius) fails to explain much. This means that the question "Is this poetry?" is often very interestingly difficult to answer. If Racter's story isn't poetry, why not? Because it's in prose? (Do we have any trouble agreeing that it's a story?) Because of its unexpected leaps of thought? Because it doesn't make *enough* leaps of thought? Because it isn't really thought at all, having been produced by a machine, unless William Chamberlain is pulling our leg, in which case, would it be satiric poetry? If it isn't poetry, where does it stop being poetry—somewhere in the middle, or at the end, or before we even read it?

The complexity of poetic interaction, the tricky dance among poet and text and reader, creates a game of hesitation. In this game a properly programmed computer has a chance to slip in some interesting moves. The brute-force effort to make a machine into a human poet seems doomed to death by boredom. A more promising attack concentrates on the open field where poems and readers meet. Readers intuitively deduce the existence (and the situation and feelings) of a poet. They do this through what is often the only evidence before them, the poem. This struggle to deduce, to interpret, leaves readers open, exposed, their lines extended, their own momentum propelling them headlong toward meaning.

Only poets, though, are likely to spot this opening. And only somebody with a modicum of programming experience (or with access to somebody who programs) is likely to find a way of exploiting it. The combination is unusual, which is why there has been so little computer poetry. As computers multiply explosively, the conjunctions are beginning to happen. Jackson Mac Low has been using a computer to automate poetic procedures for several years; Paul Hoover and others are exploring related topics. And the Internet sprouts ever more vigorous discussions of "hypertext," multiply linked on-line text.

Hypertext offers exciting possibilities as a way to present poetry,

and more than that. Like any new medium, it will change the way poets write poems and readers read them. The typewriter gave poets more precise control over spacing on the page, which they learned to use to control the meanings of words. The printing press changed poetry's audience, which changed poetry. Writing itself had complicated effects on both the nature of poetry and its place among other kinds of verbal art.

(Robert Pinsky, one of the earliest literary hypertext authors, surveyed the state of affairs as of March 19, 1995, in "The Muse in the Machine: Or, The Poetics of Zork," in the *New York Times Book Review*. Later the same year, the University of Michigan Press published Michael Joyce's *Of Two Minds: Hypertext Pedagogy and Poetics*.)

Hypertext and its cousins represent "computer poetry" in one sense of the phrase. Yet these debates about a new medium for poetry's *presentation* haven't dealt much with the use of computers in the *composition* of poems. The work of Racter and Auto-Beatnik appeared on the old-fashioned page. This is "computer poetry" in a much different sense. By actually programming the computer to help to select and manipulate words, we can probe, even more intimately than with hypertext, into the poet's and reader's relation to language. This is the kind of project that I'll be exploring here. As we'll see, the question isn't exactly whether a poet or a computer writes the poem but what kinds of collaboration might be interesting.

So this book gives a report from the front: a firsthand account of some experiments in using computers to help write poems. It's a fragment of autobiography. This can be taken partly as a disclaimer. I won't be reviewing the latest developments in artificial intelligence or current research in generative linguistics. I'm not a professional programmer. I'm a poet, which is to say a "professional" in a field where "amateur" would be a fair synonym. Above all, I'm a person who likes to work in more than one area of thought at once. (These "areas" have been defined by convention. Crossing their borders isn't a crime against nature.) I've followed some lines of research in this double area—computers and poetry—a little farther than any-

one else I know of. In the course of this work I've written several poems that I and other people have found valuable. Some poems are included in the Appendix.

Since this is autobiography, the rule is, *don't wait for final results*. In the future I may go beyond what I describe in these chapters, or my work may lead me off in new directions that have nothing to do with computers. What I offer here is a momentary cross-section: one poet's thought, at one point in history, about one corner of the art.

I'd like to thank the Ingram Merrill Foundation for a fellowship that allowed me to write the first draft of this book. Actually, the Foundation's intention was to support me while I wrote poems, and I hasten to add that I've *been* writing poems, very diligently and happily. I hope they won't mind that this book happened to come out too.