FORMAL AND COMPUTER POETRY

G. Epstein
Indiana University
Bloomington, IN 47401

B. Shneiderman
University of Maryland
College Park, MD 20742

TECHNICAL REPORT NO. 62

FORMAL AND COMPUTER POETRY

G. EPSTEIN

B. SHNEIDERMAN

APRIL, 1977

Submitted to POETICS February, 1976

Revised April, 1977

1

There has been increasing activity over the last twenty years in the investigation of formal aspects of languages. The activity is due in part to advances in our understanding of these aspects, stemming from work of N. Chomsky and other researchers, and in part to the development of formal programming languages, such as FORTRAN and ALGOL, for use with modern day computers.

Concurrently, there have been attempts to use computers for the simulation of various types of natural language behavior, from natural language translation to the simplest kind of question-answering dialogue.

An understanding of the formal aspects of a language leads beyond grammatical and syntactical considerations [2], and beyond computer applications, to other fields, such as psycholinguistics and language acquisition [7], for example. In what follows, we take a brief look at another such example -- the field of poetics.

We begin by defining a <u>formal poem</u> to be any poem which has an algorithmic or programmable structure. Thus such a poem could be written as well by a human poet as by a human programmer. It is our intention to give a number of examples for each of these. We mention that formal poems which are written by programmers and run as programs on computers to produce printed outputs are sometimes referred to as <u>computer poems</u>.

2

An easy example of formal poems is given by compact poems. A compact poem or sequence is defined to be any sequence of words where the number of words in the sequence is at least twice that of the total number of different words within that sequence. A compact poem is a member of some formal sublanguage of the formal language V^n of all finite sequences of possible permutations of these different words. Thus if there are m words in the sequence and n words in the vocabulary, then there are n^m such sequences within the formal language V^n, but only $\binom{n}{i} i^m$ different compact sequences in which exactly i different words appear (from $i=1$ up to $i = \frac{m}{2}$, so that the total number of distinct sequences in the formal sublanguage of all compact poems is:

$$\left(\sum_{i=1}^{\left\lceil \frac{m}{2} \right\rceil} \binom{n}{i} i^m \right).$$

Of course, while the formal language V^n is a finite state language, a formal sublanguage of V^n need not be.

The poetry examples (1) - (5) show that the interest in formal poems is not so much involved with an interest in the mechanical generation of computer poetry as with an interest in gaining an understanding of the semiotics for the different words in the compact poem, and of the phrase structures which contain these words. This is a human concern. A compact poem focuses attention on particular words to provide a perspective of the language which contains those words, and, of course, on the people who use that language.

3

The poetry examples (6) - (9) are all computer poems, which also serve to illustrate the limitations of mechanical generation vis-a-vis the human programmer/poet/creator. The theme throughout is that formalisms and machines can help us to appreciate our humaneness. We must only be willing to preserve our humanity rather than relinquish it.

4

Example 1 by Gertrude Stein [9] is an intensive focus on the first NO we can recall -- the NO of parents, friends, or enemies. It may be the NO which makes a YES meaningful, or it may be the NO which never allows a YES. Since when, at least.

Example 2 by Gary Snyder [8] is a game from a familiar sidewalk played on familiar states, and on the words for these states. But these are four of the United States, not abstract states.

Example 3 by the first author asks how much we can do with a single word. Here EXISTS is a word which tells us something about itself, while MUSIC is a word which makes a different kind of statement about itself.

Example 4 by R. D. Laing [3] is a circular question involving oceanic versus individual feelings, and the relation of the whole with its parts. Some of Laing's poems are actually drawn in the form of a state diagram, to emphasize the formal structure.

Example 5 is a compact poem by a child. It is a youthful plaint against blind acceptance of the adult's pronouncements. While the most naive of these poems, it seems to rely the most on feelings and their associated semantics.

It is not meant to suggest that there is no interest in compact or formal poems generated by so-called artificially intelligent computers, for these are just as much part of our

5

milieu as the examples above. It is only meant to highlight the interest in formal poems when they include a large part of the human element, and to indicate that this interest is not dependent upon computers and automata. We show next that even the interest in computer poems devolves upon these concerns, and not upon the issue of devising bigger, better, or "smarter" computers.

Example 6 is by the second author using a computer program to produce the actual expression of the poem on the printed page. The program (see Figure 1) simply creates all of the 32 sentences possible with five binary choices of sentence components. A naive reader may find some attraction to the poem, but it is hardly profound. It is close to the logical calculus based, formalist poetry that Laing develops in Knots [3].

Example 6 differs from more familiar computer generated poems which simply draw on a random selection of words to place in a syntactically rigid framework. These random poems (see examples 7,8,9) can be occasionally alluring, but the computer is never aware of the attraction. The computer is used solely as a sophisticated typewriter which can generate a large number of poems by randomly choosing words and placing them in a rigid syntactic framework. Of course, the programmer/poet/creator constrains the mood of the poems by choosing appropriate verbs or nouns (for example, erotic or religious). After a large number of poems have been produced, the programmer/poet/creator selects

6

those which make some sense and are intriguing to the human reader, and could then introduce modifications before making further computer runs. In a way, this is more computer aided poetry (CAP) than poetry or computer poetry alone.

Getting the computer to read the poems and comment on them or select the best is a task well beyond current technology (1). Petrick's intelligent technical essay on "Understanding understanding poetry" (5) delves into the difficulties of having computers "understand" poetry. He points out the relatively tractable syntactic difficulties first and then focuses on major semantic difficulties. Our capacity for expressing emotions, values, sensitivities and concerns in a formal notation is practically non-existent. In the long run we expect that we will stop trying to do so because we will recognize the uselessness of the task and the potentially counterproductive results. The fantasy of making the computer a poet must never lead to making human poets into simulated computers or calculators. As our capacity to program computers to perform increasingly complex tasks grows, we will more clearly perceive the boundary between the computer as an inanimate tool and the impressive creative skills of humans.

Computers operate at very simplistic levels. They can be programmed to select words randomly from a list and produce volumes of poems. However, computers are not programmed with the motivation and the understanding of the overall structure

7

and purpose of a poem. Computers are merely tools, whose intelligence more closely resembles that of a pencil than a person. Humans can use the computer tool to create poems, but we have not found the computer to be especially helpful in producing meaningful poetry. Example 6 has an overall structure which is described by the program. As such we might consider the program itself to be a poem and not even feel the need to have it execute and print its output. While the art behind a good program is not the art behind a good poem, these arts are allied in close ways.

Poems are not merely sequences of characters on a printed page. A poem embodies the human concerns of the poet and is intended to produce resonances in the reader. A poem is a form of communication among humans. The reader is influenced by knowledge of the poet's background and experiences a poem differently if told beforehand that it was produced by a computer. The possibility of a human relationship or communication is destroyed. Withholding information about the poem's origins circumvents the issue and does not make the output of the computer more worthwhile, should the poem be in fact the output of a computer. This Turing type of computer/human testing at most makes the computer output impressive or apparently subtle, rather than intelligent or feeling.

Still, work on generating poems by computer is useful for the occasional fresh image that is produced and recognized by the programmer/poet/creator. Possibly, we will find convenient

8

ways to describe more complex ideas in the form of a program which generates a full poem.

Rumelhart's story schemas (6) or Minsky's frames (4) offer rich matrices on which to build a meaningful poem. But no matter how elegant a poem is produced by a program which has been generated by a human programmer, the reader will be deprived of the chance for human communication and relationship which can only be obtained directly with the human programmer A poem is an opportunity for a shared human experience. Computers could be used to generate millions of poems, but it is the intelligent selection of a poem expressing human concerns that is appreciated by the reader. Using computers to generate poems is analogous to setting up a movie camera to run automatically and take millions of frames. A photographer may select ten frames for an exhibit, but it is the selection of the specific frames that we respect. At the same time, we might respect and appreciate the <u>idea</u> of setting up a movie camera to take photos at Times Square at rush hour. It is the idea of doing this which is analogous to the idea of writing a program which produces poems.

9

a no, a no since, a no since when,
a no since when since, a no since when
since a no since when since, a no since,
a no since when since, a no since, a no,
a no since a no since, a no since, a no
since.

By: Gertrude Stein

EXHIBIT 1

10

FOUR CORNERS HOPSCOTCH

"Arizona

COL orado

Utah

New MEX ico

AriZona

U TAH

Colorado &

New MEXico."

By: Gary Snyder

EXHIBIT 2

11

"Exists" exists.

""Exists" exists" exists.

"Resounds" resounds.

""Resounds" resounds" resounds.

Music

EXHIBIT 3

12

All in all
Each man in all men
All men in each man

All being in each being
Each being in all being

All in each
Each in all

By: R. D. Laing

EXHIBIT 4

13

Why do we have to

Why do we got to

Why, why, why

Why do we?

EXHIBIT 5

```
I THINK THAT YOU LOVE ME
I THINK THAT YOU LOVE YOURSELF
I THINK THAT YOU DON'T LOVE ME
I THINK THAT YOU DON'T LOVE YOURSELF
I THINK THAT I LOVE ME
I THINK THAT I LOVE YOURSELF
I THINK THAT I DON'T LOVE ME
I THINK THAT I DON'T LOVE YOURSELF
I DON'T THINK THAT YOU LOVE ME
I DON'T THINK THAT YOU LOVE YOURSELF
I DON'T THINK THAT YOU DON'T LOVE ME
I DON'T THINK THAT YOU DON'T LOVE YOURSELF
I DON'T THINK THAT I LOVE ME
I DON'T THINK THAT I LOVE YOURSELF
I DON'T THINK THAT I DON'T LOVE ME
I DON'T THINK THAT I DON'T LOVE YOURSELF
YOU THINK THAT YOU LOVE ME
YOU THINK THAT YOU LOVE YOURSELF
YOU THINK THAT YOU DON'T LOVE ME
YOU THINK THAT YOU DON'T LOVE YOURSELF
YOU THINK THAT I LOVE ME
YOU THINK THAT I LOVE YOURSELF
YOU THINK THAT I DON'T LOVE ME
YOU THINK THAT I DON'T LOVE YOURSELF
YOU DON'T THINK THAT YOU LOVE ME
YOU DON'T THINK THAT YOU LOVE YOURSELF
YOU DON'T THINK THAT YOU DON'T LOVE ME
YOU DON'T THINK THAT YOU DON'T LOVE YOURSELF
YOU DON'T THINK THAT I LOVE ME
YOU DON'T THINK THAT I LOVE YOURSELF
YOU DON'T THINK THAT I DON'T LOVE ME
YOU DON'T THINK THAT I DON'T LOVE YOURSELF

TIME:  0.56 SECS.

READY
LISTNH
10 DIM A$(2),B$(2),C$(2),D$(2),E$(2)
20 A$(1) = "I "
30 A$(2) = "YOU "
40 B$(1) = "THINK "
50 B$(2) = "DON'T THINK "
60 C$(1) = "YOU "
70 C$(2) = "I "
80 D$(1) = "LOVE "
90 D$(2) = "DON'T LOVE "
100 E$(1) = "ME"
110 E$(2) = "YOURSELF"
120 FOR I = 1 TO 2
130 FOR J = 1 TO 2
140 FOR K = 1 TO 2
150 FOR L = 1 TO 2
160 FOR M = 1 TO 2
170 PRINT A$(I)+B$(J)+"THAT "+C$(K)+D$(L)+E$(M)
180 NEXT M
190 NEXT L
200 NEXT K
210 NEXT J
220 NEXT I
230 END

READY
```

EXHIBIT 6

15

MYSELF MANIFEST

Margaret Chisman

My head thrives on pain
Unseen by guilt
Not relaxing not seducing
Comfort if controlled
Corrupts

My eye quickens with grief
Bleak with doubt yet true
Rarely hoarding rarely aching
Sorrow if withheld
Consoles

My hand delights in rejection
Eager for despair ever calm
Beyond quarrelling beyond dreaming
Friendship if followed
Refreshes

*Producing Computer Poetry
by Margaret Chisman

16

SAMPLE RUN

Lovely dumb girl
.... faltering crisp eye . . . shimmer
Dumb sudden delight

Madly swiftly whisper
.... kindly softly shimmer ray
Brutally swiftly

Girl muscular lovely
Muscular .... sky big short muscular
Trumpet short sudden

*Haiku Generator
by Paul J. Emmerich

EXHIBIT 8

17

THE STEEL

Veins rouse the steel
Joy protracts the direction

Mists utter the statues
Red fire weaves lovers

Naked pleasures begin
The steel

*Roses are red, Computers are blue
by David H. Ahl

18

References

1. R. W. Bailey, "Computer-assisted poetry: The writing machine is not for everybody", in J.L. Mitchell, ed., Computers in the Humanities, Minneapolis: University of Minnesota Press, 1974, pp. 283-295.

2. J. E. Hopcroft, J. E. Ullman, Formal Languages and their Relation to Automata, 1969, Addison-Wesley.

3. R. D. Laing, Knots, 1970, Random House.

4. Minsky, M., A framework for representing knowledge. In P. H. Winston (Ed.), The psychology of computer vision. New York: McGraw-Hill, 1975.

5. Petrick, Stanley R., Understanding Understanding Poetry, in Research Report, RC 6270, 10/29/76, IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.

6. Rumelhart, D. E. Notes on a schema for stories. In D. G. Bobrow & A. Collins (Eds.), Representation and understanding: Studies in cognitive science. New York: Academic Press, 1975.

7. D. I. Slobin, Psycholinguistics, 1971, Scott, Foresman.

8. G. Snyder, Regarding Wave, 1970, New Directions.

9. C. Van Vechten, ed., Selected Writings of Gertrude Stein, 1962, Random House.