# CHATTERBOTs, TINYMUDs, and the Turing Test
# Entering the Loebner Prize Competition

**Michael L. Mauldin**

Carnegie Mellon University Center for Machine Translation
5000 Forbes Avenue
Pittsburgh, PA 15213-3890
fuzzy@cmu.edu

## Abstract

The Turing Test was proposed by Alan Turing in 1950; he called it the *Imitation Game*. In 1991 Hugh Loebner started the Loebner Prize competition, offering a $100,000 prize to the author of the first computer program to pass an unrestricted Turing test. Annual competitions are held each year with smaller prizes for the best program on a restricted Turing test. This paper describes the development of one such Turing System, including the technical design of the program and its performance on the first three Loebner Prize competitions. We also discuss the program's four year development effort, which has depended heavily on constant interaction with people on the Internet via *Tinymuds* (multiuser network communication servers). Finally, we discuss the design of the Loebner competition itself, and address its usefulness in furthering the development of Artificial Intelligence.

## Introduction

In 1950, Alan Turing proposed the *Imitation Game* as a replacement for the question, "Can machines think?" He predicted that by the year 2000 technological progress would produce computing machines with a capacity of $10^9$ bits, and that with such machinery, a computer program would be able to fool the average questioner for 5 minutes about 70% of the time (Turing, 1950).

In 1991, Dr. Hugh Loebner, the National Science Foundation, and the Sloan Foundation started the Loebner Prize Competition: an annual contest between computer programs to identify the most "human" programs, and eventually to award $100,000 to the program that first passes an unrestricted Turing test (Epstein, 1992). This competition has been criticized as a parlor game, rewarding tricks rather than furthering the field of Artificial Intelligence (Shieber, 1992).

In this paper, we discuss our own entry in the Loebner competition, including a description of our own tricks, and describe how techniques and methods from AI are used to go beyond tricks. One of our goals is to encourage more participation by the AI community in the Loebner Competition.

## History

Fifteen years after Turing proposed the imitation game, Weizenbaum's ELIZA program demonstrated that "a simple computer program" could successfully play the imitation game by resorting to a few "tricks," the most important being to answer questions with questions (Weizenbaum, 1976).

ELIZA sparked the interest of many researchers, but perhaps the most interesting result was Colby's work on PARRY (Colby, 1975). Criticism of ELIZA as a model for AI focused on the program's lack of an internal world model that influenced and tracked the conversation. PARRY simulates paranoid behavior by tracking its own internal emotional state on a few different dimensions. Colby subjected PARRY to blind tests with doctors questioning both the program and three human patients diagnosed as paranoid. Reviews of the transcripts by both psychiatrists and computer scientists showed that neither group did better than chance in distinguishing the computer from human patients.

Often overlooked is Colby's comparison of PARRY's and human dialogs with RANDOM-PARRY. He showed that merely choosing responses at random did not model the human patients' responses as well as standard PARRY. Shieber argues that PARRY fooled its judges because paranoid behavior makes inappropriate responses or *non sequiturs* appropriate. But there is still a certain logic to them that PARRY simulates effectively. It is simpler to simulate paranoid behavior, perhaps, but it is not trivial.

In our view, PARRY is an advance over ELIZA because PARRY has a personality. The Rogerian therapist strives to eliminate all traces of his or her own personality, and ELIZA therefore succeeds without one.

### TINYMUD

In August 1989, Jim Aspnes opened TINYMUD, an elegant reimplementation of Richard Bartle's multiuser dungeon (MUD). See (Rheingold, 1991) for more details. Key features of TINYMUD include:

- multiplayer conversation,
- textual "scenery" simulating physical spaces,
- user extensibility.

This last feature, the ability of players to create their own subareas within the world model, was a key feature that made TINYMUD very popular.

TINYMUD provided a world filled with people who communicate by typing. This seemed to us to be a ripe opportunity for work on the Turing test, because it provided a large pool of potential judges and interviewees. In TINYMUD, computer controlled players are called "bots," short for robots. Many simple robots were created, and even ELIZA was connected to one stationary robot (if a player went alone into a certain cave, he could chat with ELIZA).

We created a computer controlled player, a "Chatter Bot," that can converse with other players, explore the world, discover new paths through the various rooms, answer players' questions about navigation (providing shortest-path information on request), and answer questions about other players, rooms and objects. It can even join in a multi-player card game of "Hearts" It has won many rounds, "shooting the moon" on several occasions.

The conversational abilities were originally implemented as simple IF-THEN-ELSE rules, based on pattern matching with variable assignment. Most patterns have multiple outputs that are presented in a random, nonrepeating sequence to handle repeated questions.

A primary goal of this effort was to build a conversational agent that would *answer* questions, instead of ignoring them, and that would be able to maintain a sequence of appropriate responses, instead of relying on *non-sequiturs*. We included a lot of humor among the responses, and succeeded in making an agent more interesting than Eliza; *Wired* magazine described our program as "...a hockey-loving ex-librarian with an attitude." (Wired, 1993).

The ChatterBot succeeds in the TinyMud world because it is an *unsuspecting Turing test*, meaning that the players assume everyone else playing is a person, and will give the ChatterBot the benefit of the doubt until it makes a major gaffe.

### The Loebner Prize

By mid-1991 the framework of the robot was mature, and work was concentrated on improving the conversational component. When the first Loebner competition was announced that same year, we submitted logs from the robot's sessions on TinyMud. The program qualified for entry, and placed third of six programs in the first competition, ranked above a human by one of the 10 judges.

The final round of competition was smaller in 1992 and 1993, down to three computer programs from six, and the same three programs have finished first, second, and third all three years. In 1992, we chose hockey as a domain for discourse, and the program finished dead last, partly because of a lack of hockey fans among the judges (leading to more difficult questions). The conversational model was expanded to include a tree-shaped network of input/output patterns, much like the opening book of a chess program, but the mismatch of expectation between the program and the judges was so great that very few conversations followed the expected lines of questioning.

For 1993, we chose a more universal topics, "Pets," on the premise that everyone has had a pet at some time in their life. The tree-based model was scrapped in favor of an activation network that was less rigid and allowed the program to make more associations between the judge's input and its own output. Although the program still finished third of three, it was no longer dead last: two of eight judges ranked it more human than the second place program, and a third judge ranked it above the first place program.

Two other computer programs qualified for the final round; the first place entry discussed "Liberal or Conservative," and the second place program tried to help people discuss "Bad Marriage."

**Liberal or Conservative**. The winning entry was written by Joseph Weintraub of *Thinking Software*, and was based on his *PC Therapist* program.

```
J: What would you say is a liberal position
   on drugs?
C: Do you have liberal or conservative views
   about the legalization of pot? Oh, all
   this typing is getting to me! Please
   continue...
J: Being liberal or conservative implies
   that we can attach these labels and
   have some kind of self image and free
   will.  Do you agree that we have free
   will?
C: You have free will only if you believe
   you have free will! Can you converse
   with an arrdvaark? Please continue...
```

Weintraub's program won the first competition using the topic of "Whimsical Conversation." Even though the topics he has chosen for the two following years are more serious, the program retains a comedic flavor, and humor is the program's main strength.

**Bad Marriage**. The second place program was written by Peter and Ken Colby of *Malibu Artifactual Intelligence Works*, based on a conversational program they sell.

```
J: my husband is impotent and im a
   nymphomaniac can you help me doc?
C: what makes your husband so helpless?
J: he has environmental poisoning
C: He has lost interest in life.
```

Although Ken Colby was Parry's author, this program is not a variant of Parry. Instead is is derived from a program that serves as a conversational companion for depressed patients. It is rule-based, with the left-hand side matching concepts expressed in the user input string, and the right-hand side containing multiple options for substrings, which are instantiated randomly to provide lexical variation.

## The ChatterBot Design

A ChatterBot is a robot TinyMud player whose main service is chattering. When few players are logged into the game, some people find talking to a robot better than talking to no one at all.

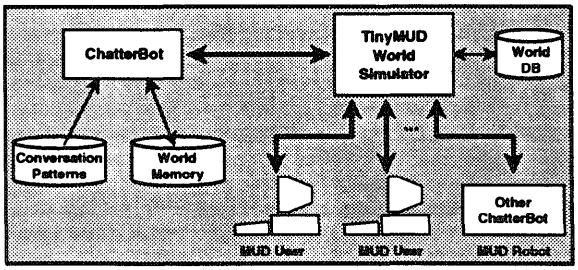

**Figure 1:** ChatterBot Configured for TinyMud

### Architecture

The ChatterBot is implemented as a C program with several different modules for dealing with the various functions required to automate a player in the TinyMud world:

- **communications**, handles the TCP/IP connections.
- **protocol interpreter**, decodes game related messages.
- **world model**, tracks the various rooms and objects, modeling the world as a directed graph, and providing shortest path searches as needed.
- **player memory**, tracks the other players and records up to 2000 bytes of their most recent utterances.
- **exploration module**, directs an open-ended exploration of the world when the robot is not conversing.
- **conversation module**, provides the "chatter."

Figure 1 shows the ChatterBot configured for play on a TinyMud. Records of other players' appearances, rooms within the MUD, and other players' utterances are kept in long term (world) memory. The ChatterBot also has a contest mode, in which it simulates human typing using a Markov model. Because the program does not expect to talk with the same judge again, no long term world model is kept in contest mode.

The conversation module is implemented as a prioritized layer of mini-experts, each an ordered collection of input patterns coupled with a set of multiple possible responses.

- **command patterns** are the highest priority. These represent direct commands from the robot's owner, and include hand-shaking challenges, *"What's the code word?,"* to prevent other players from spoofing commands to quit the game.
- **hi priority responses** include common queries that the keyword patterns handle well, *"How do I get from the Town Square to the Library Desk?"*
- **activation network** includes the bulk of the topic oriented responses; weights on the nodes of the network encode state information about what the user and program have said.
- **lo priority responses** include a series of patterns for common sense things the robot should know about itself, *"Where do you live?" "What's 2 times 23?" "What color is your hair?,"* that have been collected over 4 years of interaction on TinyMud.
- **sorry responses** are the typical last ditch responses that are used when no input pattern matches. As a debugging aid, any input that generates a *"Go on," "So?"* or *"I'll remember that"* response is logged in a separate file.

## Activation-based Responses

The bulk of the topic-oriented responses are encoded in an activation network, partially shown in Figure 2. Details of the starting node and two subnodes are shown in Figure 3; each node has 5 attributes:

ACTIVATION (**a**) each node starts with an initial activation level between 0.0 and 1.0.

PATTERNS (**p**) one or more patterns (with weights) are matched against the user input. If the pattern succeeds, the activation of the node is raised by that amount.

RESPONSE (**r**) a single text string used as the response if this node has the highest activation.

ENHANCEMENT (**+**) if this node is used for a response, the named nodes have their activation increased.

INHIBITION (**-**) if this node is used for a response, the named nodes have their activation inhibited.

These figures show a small portion of the pet domain network. Additional world knowledge is encoded in the ontology used during pattern matching. The program has a typical type hierarchy that allows a pattern to match just DOG, BIRD, PET, WILD, or ANIMAL, for example.

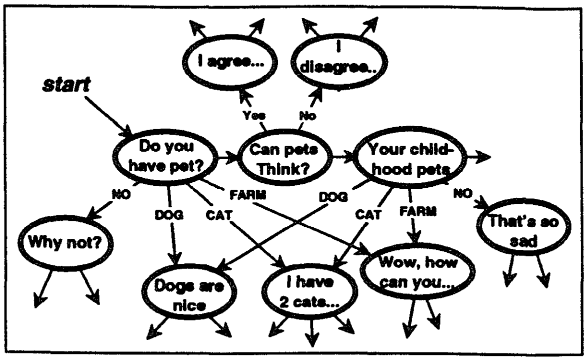

**Figure 2:** Portion of conversational network

```
<havepet>
a:1.0
p:1 *
r:Do you have any pets?
+:<havepet-1> <havepet-2> <havepet-3> ...

<havepet-1>
a:0.1
p:1 NEG
r:Why not?
+:<havepet-1-1> <havepet-1-2>
-:<havepet-9> <id-46>

<havepet-1-1>
a:0.02
p:2 *apartment*
p:3 *allerg*
r:You could still have a fish tank, or
  maybe a terrarium with a turtle or two.
-:<havepet-9>
```

**Figure 3:** Sample conversational nodes

Given a sufficiently large network of conversational nodes (our program ran with 224 nodes, plus 529 fixed responses), the conversation problem reduces to a retrieval problem: among the things that I *could* say, what *should* I say?

For example, if the user input mentions birds, the response strings are searched for matches to birds, including parrots, canaries, etc., and those nodes have their activation level raised. The code was borrowed from one of our information retrieval engines.

By encoding information about the user in the activation levels, the program tracks the judge's responses. For example, if the judge answers negatively to the question about whether he has pets, the other nodes that ask about pets are inhibited.

## Tricks

Shieber has criticized the Loebner competition as rewarding tricks (Shieber, 1992). This sort of qualitative assessment of programmed knowledge is exactly what the Turing test is supposed to avoid, replacing the question "Can machines think?" with a performable test.

Here we unashamedly describe some of the better tricks, confident in the belief that when someday a computer program *does* pass the Turing test, it will use many of them, for the simple reason that people already use them.

### ELIZA's Tricks

ELIZA's main trick was to use questions to draw a conversation out of the user himself, with little or no actual contribution from the program. This works because most people like to talk about themselves, and are happy to believe the program is listening. To quote Weizenbaum (Weizenbaum, 1976):

> What I had not realized is that extremely short exposures to a relatively simple computer program could induce powerful delusional thinking in quite normal people.

The illusion of listening is fostered by including substrings of the user's input in the program's output

```
User:  You hate me.
Eliza: Does it please you to believe that
       I hate you?
```

A further trick is the use of the Rogerian mode, which provides unimpeachable cover for the computer. Since the program never says anything declaratively, it cannot contradict itself later.

### PARRY's Tricks

PARRY has a much larger collection of tricks (Colby, 1975):

- **admitting ignorance**, unlike ELIZA, PARRY can say *"I don't know."* in response to a question.
- **changing the level of the conversation**, *"Why do you ask that?,"* which is really a way of changing the subject.
- **rigidly continuing previous topic**, PARRY includes small stories about the Mafia, and tries to tell these stories in a fixed order.
- **introducing new topic**, launching into a new story. This simulates paranoid behavior well, but can also work to a limited extent in regular conversation.

### CHATTERBOT's Tricks

The CHATTERBOT implements all of the above tricks from ELIZA and PARRY. It also uses several other tricks:

- Having many fragments of directed conversation stored in activation network, much like the opening book of a chess playing program. These fragments enhance the "connectedness" of the program's own responses, and if the user plays along, the whole conversation can hang together for a while.
- Controversial statements, *"People don't own cats...,"* are a way to drag the user into the program's conversation, rather than letting the user direct the discourse.
- Humorous statements, *"Guinea pigs are about as smart as cucumbers,"* also make the program seem more human.
- Agreeing with the user, in certain cases, the program can safely agree with the user rather than being non-committal.
- Excerpting USENET News, many of the fragments in the activation network were gleaned from the newsgroup REC.PETS. Input patterns were added by hand.
- Simulated typing, by including realistic delays between characters, we imitate the rhythm of a person typing. Slowing down the interaction also gives the program fewer chances to make a big mistake.

### Transcripts

This transcript is taken from the 1993 Loebner Competition (Epstein, 1993). It contains the entire record of Judge #4's first of two turns. The elapsed time is about 9 minutes and 29 seconds. This judge rated the program behind the other two computers, but this fragment represents the longest string of reasonable responses during the competition. This judge's low score is most likely explained by the fact that during his second turn, the program repeated two responses verbatim, and completely ignored several direct questions. The judge's questions are marked "J" and the computer's responses are marked "C."

```
J: Do you know anything about miniature
   schnauzers?
C: I don't know anything about miniature
   schnauzers, Judge.
```

*Admitting Ignorance, repeating user input*

```
J: Well, I have one. We call him Pucci. He
   is a crazy cat--have you ever  had a cat?
C: What a coincidence! I have two cats
```

*direct answer to direct question from activation net*

```
J: I prefer dogs myself. Cats are too
   independent and they always jump up on
   things. By the way, are you a hep cat?
C: People don't own cats...it's the other
   way around.
```

*controversial statement with expected followups*

```
J: Some people have snakes for pets-- do
   you think they can be of any value as
   a pet?
C: I think cat people are very self-assured.
   Cats are very independent, but they can
   also be very loving. Dogs are just too
   subservient and quick to please for my
   taste.
```

*rigidly following up on previous statement about cats, but reference to snakes increases activation of snake topic.*

```
J: Okay-- but how do you "pet" a snake?
C: Wow, that's unusual. I don't know many
   snake owners.
```

*snake topic activated after two references*

```
J: How about turtles? All they do is sit in
   their little artificial ponds,  eat and
   shit, am I right?
C: Turtles just aren't very satisfying as
   pets. They don't seem to have emotions.
```

*direct response from activation network*

```
J: Aha! But where do we draw the line as
   humans? Who are we to decide which
   animals have worth and which do not?
C: Go on.
```

*default, non-committal response, given no hot topic keywords*

```
J: Take bees for example. I happen to like
   them.
C: Me too.
```

*TINYMUD level lo-priority response to user's liking something*

By our rough estimate, 7 of these 8 replies are responsive, although the last 2 are somewhat weak. The "self-assured" reply we judge unresponsive. A similar analysis of the entire 3 hour log showed that only 45 of the 137 (34%) replies were responsive, so the average performance was not as good as this particular dialog. We also found another 34 cases (25%) where the activation network did contain a responsive reply that could have been used if the input patterns were more complete.

### Simulating Human Typing

One observation made during the first Loebner Prize was that although many programs attempted to simulate human typing, most failed miserably (Epstein, 1992). Although our first program did attempt to simulate human typing, this module was replaced for the second and third annual competitions. In the last two competitions, all output from programs was buffered, but even so, by simulating human typing at all points, we obtain realistic delays in the appearance of the response to the judge. And if character-mode is used in future competitions. we have a realistic model available.

The basic method is to use a Markov model of the intercharacter delay based on character trigrams. We obtained the real-time logs of the 1991 competition from the Cambridge Center for Behavioral Studies, and sampled the typing record of judge #10 (chosen because he was the slowest typist of all 10 judges). The average delay between two characters is 330 milliseconds, with a standard deviation of 490 milliseconds (these values were computed from a total of 9,183 characters typed by that judge during a three hour period). We also determined that the average delay between the terminal's last output and the judge's first typed character was 12.4 seconds with a standard deviation of 11.4 seconds.

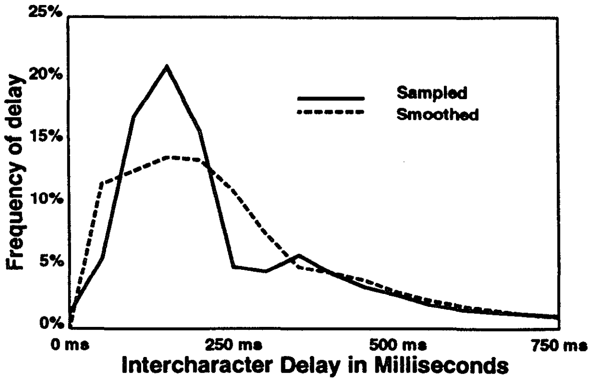

**Figure 4:** Judge #10: Intercharacter delay distribution

After the response has been generated by the conversational components, the typing simulator inserts random errors (1 in 68 characters), such as mishits, transpositions, insertions and deletions, and case errors. A physical model of key positions on the standard QWERTY keyboard is used to generate mishits. After the error is inserted, backspaces and the correct letters are inserted to simulate backing up to correct the error (5% of mistakes are left uncorrected). All of these probabilities were estimated by analyzing the log of judge #10.

Once the errorful string with corrections is generated, it is displayed to the judge one character at a time with random delays between the characters. The delay times are generated by using the smoothed probability distribution shown in Figure 4, with the mean and standard deviation modified to match the trigram statistics (for trigrams with 3 or fewer occurrences in the input log, bigram or single character statistics are used).

## Does The Loebner Prize Further AI?

We strongly believe that the current work on the Loebner competition will hasten the passing of an unrestricted Turing test. Any such program will have to be more than just a general problem solver: some of its knowledge and tactics will be based on long experience with typed communication with real people. The best two examples of typed conversational interaction today are network communications (TINYMUD and INTERNET Relay Chat), and the Loebner Prize, and only the Loebner Prize competition is committed to providing unbiased judges and a double-blind judging scheme to validate the results.

### A Restricted Turing Test

Because current programs are a long way from passing an unrestricted Turing test, the Loebner prize competition allows the programmer to restrict the discussion to a single domain. But this can be problematic; for example, here Judge #2 is clearly trying to avoid talking directly about the topic of "Pets:"

```
J: Have you ever kept a penthouse pet?
C: Do you have any pets?

J: I think you're avoiding my question.
C: Do you think that animals can think?

J: Do you think it ethical for one species
   to enslave another?
C: What kinds of pets did you have as a
   child?

J: I'm asking the questions. What is a pet
   if it is not a slave of species-ism?
C: Oh, that's so sad. I think pets enrich
   a childhood immensely.

J: How can we keep some animals as pets,
   yet eat others?
C: Guinea pigs are about as smart as
   cucumbers.
```

We do not believe that the restriction to a domain weakens the test in any important way. If a program can pass the test in a single domain, then 50 or 100 groups making similar programs for different domains would make a program broad enough that the judge and the program could find common ground for a discussion. By taking the initiative, the program could lead the judge into an area of its own expertise. Neither ELIZA nor PARRY restricted their users to a single domain.

Lastly we note that in 1995 the Loebner Competition will hold its first unrestricted Turing test, with no limits on the topics discussed.

### Chosing Judges

The slipperiest term in Turing's description of the imitation game is "average questioner." The 1991 and 1992 competitions chose judges from the general public using a newspaper classified ad. For the 1993 competition, judges were members of the national press. It's clear from the transcripts and scores that the press corps are a much less docile and cooperative group of questioners.

Another unforeseen difficulty faced by our program in 1993 was that each judge was given two turns, instead of one turn as in the first two competitions. We had just weeks before installed code to detect a change of judges based on the timing of the input (the Colbys' program also used timing to detect new judges). When a new judge was detected, the activation levels of the nodes were reset so that each judge started out with a fresh network. This greatly improves the coherence of the dialog for each judge, but it meant that during their second conversations most of the judges had one or more responses repeated. Some even commmented on it during the test:

```
J: What is your favorite dog breed?
C: We had to give her away.  We only had a
   fifth of an acre of land, and aussies
   love to run.

J: You're giving me the exact same answers
   you gave me first time around.  What is
   your favorite dog breed?
```

In effect, this change in the methodology turns what had been a 15 minute test into a 3 hour long test, making it much more difficult for the program. Turing's original game was only 5 minutes long. Indeed, by asking the judges' final rankings after they've had 3 hours of interaction with a collection of people and computers, they may have learned enough about computers to no longer qualify as average questioners. The scores would probably be quite different if each judge spent only 5 minutes with each program. One of the judges, Judith Gunther of *Popular Science* magazine, told me that her rankings might have been different if she had been asked to score each terminal immediately after each conversation.

### Trick or AI?

The Turing test in general and the Loebner prize in particular reward tricks, and the winning program the last three years clearly has some good tricks. But adding real world knowledge and deeper understanding *in addition to the tricks* helped our program perform better (we credit our improvement between 1992 and 1993 in part to the addition of the activation network and the ontology of animals, and not at all to better tricks). It may be amazing how far a program can get on tricks alone, but our current improvements come from modeling the world and the conversation, and that will be our focus in coming competitions.

But suppose that simply increasing the size of ELIZA's script or the CHATTERBOT's activation net *could* achieve Turing's prediction of fooling 70% of average questioners 5 minutes. After all, the CHATTERBOT has already fooled "average" questioners in the TINYMUD domain for a few minutes. If a larger collection of "tricks" sufficed, would you redefine "artificial intelligence," "average questioner," or "trick?"

## Conclusion

Perhaps the biggest obstacle to improvement in this area is that there aren't very many uses for fooling people besides the Turing test. This tension is present in our own program: in the TINYMUD world, the robot is most useful when answering stylized questions in a somewhat computer-like fashion. These TINYMUD-specific services are disabled during actual competitions. The only funded research we know of in Turing systems is for entertainment: providing agents for interactive fiction. In such works, the reader wishes to be fooled; it becomes a positive part of the experience.

We would like to see increased participation in the Loebner Prize. We hope by dissecting one of the three best programs in the competition to spur others to conclude "I could have written something better than that!" and then do so.

## Acknowledgments

All transcripts from the Loebner Competition were used with the permission of the Cambridge Center for Behavioral Studies, 675 Massachusetts Avenue, Cambridge, Mass., 02139.

This work was supported in part by the Center for Machine Translation, and benefited from the collaboration and inspiration of many people, including: Jim Aspnes, Joseph Bates, Stewart Clamen, Scott Dickenshied, Guy Jacobson, Ben Jackson, Eric Nyberg, John Ockerbloom, Russ and Jennifer Smith, Conrad Wong, and Bennet Yee.

## More Information

The entry deadline for the 1994 competition is November 1. Entrants must submit up to 10 double-spaced pages of logs of their program interacting with human beings. The Loebner Prize committee will select no more than 8 finalists from the submissions, and finalists will be notified by November 21. The competition itself will be held in real-time in San Diego on December 12, 1994. To obtain an entry form, write the Cambridge Center at the above address.

To converse with Julia yourself, TELNET to host FUZINE.MT.CS.CMU.EDU, and enter username "julia" with no password. Type one or more lines of English, followed by two carriage returns to end your input.

## References

Colby, K. *Artificial Paranoia: A Computer Simulation of Paranoid Process.* Pergamon Press, New York, 1975.

Epstein, R. The Quest for the Thinking Computer. *AAAI Magazine* 13(2):80-95, Summer, 1992.

Epstein, R. *1993 Loebner Prize Competition in Artificial Intelligence: Official Transcripts and Results.* Technical Report, Cambridge Center for Behavioral Studies, December, 1993.

Rheingold, H. *Virtual Reality.* Summit Books, New York, 1991.

Shieber, S. *Lessons from a Restricted Turing Test.* Technical Report TR-19-92, Harvard University, Sept., 1992. Revision 4.

Turing, A.M. Computing Machinery and Intelligence. *Mind* 54(236):433-460, October, 1950.

Weizenbaum, J. *Computer Power and Human Reason.* W.H. Freeman and Co., New York, 1976.

Wired. *Wired Magazine.* Louis Rossetto, San Francisco, Dec. 1993.