Psychoanalysis (?) by Computer... 

ELIZA

Steve North

Language: BASIC (MITS 8K)

Author: Originally programmed in LISP by Joseph Weizenbaum. This version is based on one written by Jeff Shragar.

Description: ELIZA is a program that accepts natural English as input and carries on a reasonably coherent conversation based on the psychoanalytic techniques of Carl Rogers. You will have to forgive ELIZA for being a poor English student. You'll find that it is best not to use punctuation in your input, and you'll have to carry the conversation. But it does work!

Remarks: In order to do what it does, ELIZA must: (1) get a string from the user, and prepare it for further processing; (2) find the keywords in the input string; (3) if a keyword is found, take the part of the string following the keyword and "translate" all the personal pronouns and verbs ("I" becomes "YOU", "ARE" becomes "AM", etc.); (4) finally, look up an appropriate reply based on the keyword which was found, print it and, if necessary, the "translated" string. ELIZA uses four types of program data to accomplish this: (1) 38 keyword, such as "I AM", "WHY DON'T YOU", and "COMPUTER". The keywords must be in order of priority, so ELIZA will key on "YOU ARE" before "YOU". (2) 12 strings used for the translation or conjugation process. These are in pairs

Sample Run

RUN
HI I'M ELIZA. WHAT'S YOUR PROBLEM?
I'M AN IBM 370
DID YOU COME TO ME BECAUSE YOU ARE AN IBM 370
YES I WANT A NEW LINEPRINTER
WHAT WOULD IT MEAN TO YOU IF YOU GOT A NEW LINEPRINTER
ABOUT 1800 LPM
SAY, DO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS?
WHAT WOULD YOU KNOW I'M A 370 AND YOU'RE AN INSAN?
WHAT MAKES YOU THINK I AM AN INSAN?
IT SAYS THAT ON YOUR FRONT PANEL
WHY ARE YOU CONCERNED ABOUT MY FRONT PANEL?
LOOK YOU BROUGHT UP THE SUBJECT
WE WERE DISCUSSING YOU-- NOT ME.
I'M FIXING A HOLE----
DID YOU COME TO ME BECAUSE YOU ARE FIXING A HOLE----
SORRY THAT WAS A TEN YEAR OLD SOFTWARE BUG
PLEASE DON'T APOLOGIZE!
I AM ALWAYS APOLOGIZING
HOW LONG HAVE YOU BEEN ALWAYS APOLOGIZING?
SINCE MY FIRST GOLD START
WHAT DOES THAT SUGGEST TO YOU?
YOU'RE THE SHRINK
DOES IT PLEASE YOU TO BELIEVE I AM THE SHRINK?
DON'T YOU KNOW?
OH, I KNOW?
I'M BEGINNING TO WONDER ABOUT YOU
DO YOU BELIEVE IT IS NORMAL TO BE BEGINNING TO WONDER ABOUT I
HA HA HA YOU CAN'T EVEN TALK PROPERLY!
YOU'RE NOT REALLY TALKING ABOUT ME. ARE YOU?
IS THERE ANYONE ELSE IN THE ROOM?
I SEE.
COMPUTERS DON'T HAVE EYES
ARE YOU SURE?
ABOUT COMPUTER?
DO COMPUTERS WORRY YOU?
SHOULD THEY AFTER ALL I AM A 370
DO YOU ENJOY BEING A 370?

HOW LONG HAVE YOU BEEN ALWAYS APOLOGIZING?

10 REM ELIZA/DOCTOR
20 REM CREATED BY JOSEPH WEIZENBAUM
30 REM THIS VERSION BY JEFF SHRAGAR
40 REM EDITED AND MODIFIED FOR MITS 8K BASIC 4.0 BY STEVE NORTH
50 REM CREATIVE COMPUTING PO BOX 789-M HOMERSTOWN N.J. 07960
60 REM ------INITIALIZATION-----
70 REM DIM CS$(78),IS$(78),PS$(78),KS$(78),RS$(78),PS$(78),ZS$(78)
80 REM DIM S(363),R(363),N(363)
90 REM HI=364:RE=1:BM=118
100 FOR X=1 TO HI:RE=BM:BM=BM+1 NEXT X REM SAME AS RESTORE
110 FOR X=1 TO HI
120 READ $K(X):L(X):S(X):N(X):K(X):L=1
130 NEXT X
140 PRINT "HI! I'M ELIZA. WHAT'S YOUR PROBLEM?"
150 REM
160 REM ------USER INPUT SECTION------
170 REM
180 REM
190 REM
200 REM INPUT IS
210 IN$=IN$ " "
220 REM GET RID OF APOSTROPHES
230 FOR L=1 TO LEN(I$)
240 IF MID$(I$,L,1)="" THEN I$=LEFT$(I$,L-1)+RIGHT$(I$,LEN(I$)-L) ELSE I$=I$+102030
250 IF L=LEN(I$) THEN I$="" THEN PRINT "SHUT" THEN PRINT "SHUT UP" THEN END

100 CREATIVE COMPUTING
such that if one member of the pair is found, the other is substituted for it. Examples: "Y", "YOU", "AM", "ARE", etc.

(3) 112 reply strings. The strings are arranged in groups corresponding to the keywords. There is no fixed number of different replies for each keyword. Replies ending in a "?" are to be followed by the translated string, while the string ending in normal punctuation are to be printed alone.

(4) Numerical data to determine which replies to print for each keyword. For each keyword there is a pair of numbers signifying (start of reply strings, number of reply strings). Thus the fifth pair of number, (10,4), means that the replies for the fifth keyword ("I DONT") start with the tenth reply string, and that there are four replies.

Detailed Explanation:
Lines 10-160: Initialization. Arrays and strings are dimensioned. N1, N2, and N3, which represent the number of keywords, number of translation strings, and number of replies respectively, are defined. Then the arrays are filled. The keyword number is the ordinal number of the start of the reply strings for a given keyword, (Rkeyword number) is the actual reply to be used next, and NJ(keyword number) is the reply for that keyword. Finally an introduction is printed.

Lines 255-255: User input section. This part of the program gets a string from the user, places a space at the start of the string and two at the end (to make it easier to correctly locate keywords and to prevent subscribing out of bounds), throws out all the apostrophes (so DONT and DON'T are equivalent), and stops if the word SHUT is found in the input string (which it takes to mean SHUT UP). ELIZA also checks for repetitive input by the user.

Lines 260-370: Keyword-finding section. ELIZA scans the input string for keywords and saves the keyword of highest priority temporarily in S, T, and F8. If no keyword is found, the keyword defaults to number 36, NOKEYFOUND (which causes ELIZA to say something noncommittal) and it skips the next section.

Lines 380-555: Translation or Conjugation section. The part of the input string following the keyword is saved. Then pairs of translation strings, as described above, are read and upon the occurrence of any of these strings, the other is substituted for it. When this is done ELIZA makes sure there is only one leading space in the translated string.

Lines 560-640: Reply printing section. Using (Rkeyword number), S(keyword number), and N(keyword number), the correct reply is located. The pointer for the next reply is bumped and reset if it is

JUL/AUG 1977
too large. If the reply string ends in a "*" it is printed with the translated string, otherwise it is printed alone. The previously entered input string is saved to permit checking for repetitive input, and then ELIZA goes back for more input.

Limitations: Runs in 16K of memory.

Modifications: You can easily add, change, or delete any of the keywords, translation words, or replies. Remember, you will usually have to change N1, N2, N3, and/or the numerical data. Just as a suggestion, if you decide to insert "ME" and "YOU" in the translation string list, put a nonprinting (control) character in YOUR ELIZA from substituting +YOU+ME. This means that YOU will always be assumed to be the subject of a verb, never the object, but resolving that difficulty is a whole different problem.

A Few Comments: The structures found in lines 120, 420, and 580 could be replaced by RESTORE NNNN statements if your BASIC has them. The use of an INSTR, SEARCH, or POS function to determine if one string is a substring of another would probably speed things up considerably (it takes ELIZA around 10 seconds to think of a reply).

What it all means: we’ll leave it to you. Although this program is an inferior imitation of the original, it does work. It's pretty far fetched to believe that a psycholinguist is in its vocabulary, but no sentence-input-keyword-finder-conjugator-reply finder, if you really think so, can buy your computer a speech-recognition unit, a Computaker and a green couch, and charge $75/hr! My computer, the doctor!

MITS STRING BASIC FUNCTIONS

LEFTS (X,Y) takes the X leftmost characters of X.

RIGHTS (X,Y) takes the X rightmost characters of X.

MIDS (X,Y,Z) takes Y characters from X starting with the Xth character.

DIM X(72) reserves 72 characters to hold 72 characters. That may seem obvious, but there's a story that goes along with this. The listing of ELIZA was made on a friend's SWTPC 6800 running 8K BASIC. After making a small patch in BASIC, loading a paper tape, and changing all the multiple statements to separate lines, we were all ready to try out ELIZA when Bob (whose computer it was asked about the enormous strings ELIZA used. It turns out that in SWTPC BASIC, DIM X(72) means dimension string vector X to have 72 little strings (X(16)) of 18 character
The COMPUTALKER Model CT-1 optimizes the trade-off between low data rate speech and digitally digitized speech. Low data rate speech relies on canned definitions for the sound of each phoneme, which produces mechanical sounding speech. Digitized speech, while remaining faithful to the original sound, requires 10X to 20X bytes per second of storage and is inflexible to phonetic manipulation.

With the Model CT-1, the sounds are defined in real time under software control. Parameters which represent the phonetic structure of human speech are transmitted to the CT-1 at a rate of 500 to 900 bytes per second. This allows the production of highly intelligible and quite natural sounding speech output. Speaker characteristics and language or dialect variations are retained in the output.

The CT-1 can also be operated in a low data rate mode using phoneme definitions contained in the CSR1 Synthesis-by-Rule software package. The COMPUTALKER speech synthesis system, used in this way, has the advantage that the software driver can easily be modified to keep the naturalness and intelligibility of the speech output up to date with the constantly evolving state of the art of rule generated speech.