

AN ADAPTIVE QUERY SYSTEM

E. Kellerman  
 IBM Corporation  
 Systems Development Division  
 P. O. Box 6  
 Endicott, New York 13670

This paper will describe an adaptive query program coded in APL. The purpose of the program is to allow users to ask questions in everyday English and to receive answers with minimal delay.

The program is "taught" the correct answers by a human "instructor", sitting at a terminal, asking it questions on the subject of interest. As the program learns the answers to some of the questions, it attempts to guess at the answer to "similar" questions. If the program is "very" sure of the answer it has given, then it does not request verification, otherwise it requests confirmation of the correctness of the answer it has given. Even when the program does not request verification of the answer given, the instructor still has the option of informing the program that an incorrect reply was given, by entering '?'. If the program gives an incorrect reply, but the instructor feels that the program should know the answer, he can request the program to 'try again'. Consider the following terminal session, which starts with a complete unknowledgeable program. The lines typed by the terminal are preceded by '\*\*\*':

HOW DO I GET OFF?  
 \*\*\*WHAT IS ANS ?  
 HIT CARRIAGE RETURN!!

WHAT PROGRAMS ARE AVAILABLE?  
 \*\*\*WHAT IS ANS ?  
 LIST, DUP, AND INTER.

WHAT DOES LIST DO?  
 \*\*\*WHAT IS ANS ?  
 LISTS CARDS.

WHAT DOES DUP DO?  
 \*\*\*LISTS CARDS.  
 \*\*\* WAS THAT RIGHT?  
 NO  
 \*\*\*WHAT IS ANS ?  
 DUPLICATES CARDS.

WHAT DOES INTER DO?  
 \*\*\*LISTS CARDS.  
 \*\*\* WAS THAT RIGHT?  
 NO  
 \*\*\*WHAT IS ANS ?  
 INTERPRET CARDS.

HOW DO I LIST CARDS?  
 \*\*\*WHAT IS ANS ?  
 USE LIST.

HOW DO I DUPLICATE CARDS?  
 \*\*\*USE LIST.  
 \*\*\* WAS THAT RIGHT?  
 NO  
 \*\*\*WHAT IS ANS ?  
 USE DUP.

HOW DO I INTERPRET CARDS?  
 \*\*\*USE DUP.  
 \*\*\* WAS THAT RIGHT?  
 NO  
 \*\*\*WHAT IS ANS ?  
 USE INTER.

HOW DO I DUP CARDS?  
 \*\*\*USE LIST.  
 \*\*\* WAS THAT RIGHT?  
 TRY AGAIN  
 \*\*\*USE DUP  
 \*\*\* WAS THAT RIGHT?  
 YES

HOW DO I DUP CARDS?  
 \*\*\*USE DUP  
 \*\*\* WAS THAT RIGHT?  
 YEP

HOW DO I INTER CARDS  
 \*\*\*USE LIST.

?  
 \*\*\* WAS THAT RIGHT?  
 TRY AGAIN  
 \*\*\*USE INTER.  
 \*\*\* WAS THAT RIGHT?  
 YEAH

HOW DO I INTER. CRDS?  
 \*\*\*USE INTER.

HOW DO I LIST CDS.  
 \*\*\*USE LIST.

WHAT PROGRAMX AVAILABLE?  
 \*\*\*LIST, DUP, AND INTER.

WHAT DOS DUP DO?  
 \*\*\*DUPLICATES CARDS.

HOW DO I DUPLIMATE CRDS.?  
 \*\*\*USE DUP.

HOW DO I INTERPRETARE CRDS.?  
 \*\*\*USE INTER.

WHAT DO INTER DO  
 \*\*\*INTERPRET CARDS.

HOW DO I LST CRDS?  
\*\*\*USE LIST.

WHAT DOES LIST DO????  
\*\*\*LISTS CARDS.

WHAT DEOS INTRE DO?  
\*\*\*INTERPRET CARDS.

HOW DO I SIGN OFF.  
\*\*\*HIT CARRIAGE RETURN!!

HOW DO I GET OFF?  
\*\*\*WHAT IS ANS ?  
HIT CARRIAGE RETURN!!

WHAT PROGRAMS ARE AVAILABLE?  
\*\*\*WHAT IS ANS ?  
LIST, DUP, AND INTER.

WHAT DOES LIST DO?  
\*\*\*WHAT IS ANS ?  
LISTS CARDS.

WHAT DOES DUP DO?  
\*\*\*LISTS CARDS.  
\*\*\* WAS THAT RIGHT?  
NO  
\*\*\*WHAT IS ANS ?  
DUPLICATES CARDS.

WHAT DOES INTER DO?  
\*\*\*LISTS CARDS.  
\*\*\* WAS THAT RIGHT?  
NO  
\*\*\*WHAT IS ANS ?  
INTERPRET CARDS.

HOW DO I LIST CARDS?  
\*\*\*WHAT IS ANS ?  
USE LIST.

HOW DO I DUPLICATE CARDS?  
\*\*\*USE LIST.  
\*\*\* WAS THAT RIGHT?  
NO  
\*\*\*WHAT IS ANS ?  
USE DUP.

HOW DO I INTERPRET CARDS?  
\*\*\*USE DUP.  
\*\*\* WAS THAT RIGHT?  
NO  
\*\*\*WHAT IS ANS ?  
USE INTER.

HOW DO I DUP CARDS?  
\*\*\*USE LIST.  
\*\*\* WAS THAT RIGHT?  
TRY AGAIN  
\*\*\*USE DUP.  
\*\*\* WAS THAT RIGHT?  
YES

HOW DO I DUP CARDS?  
\*\*\*USE DUF.  
\*\*\* WAS THAT RIGHT?  
YEP

HOW DO I INTER CARDS  
\*\*\*USE LIST.

?  
\*\*\* WAS THAT RIGHT?  
TRY AGAIN  
\*\*\*USE INTER.  
\*\*\* WAS THAT RIGHT?  
YEAH

HOW DO I INTER. CRDS?  
\*\*\*USE INTER.

HOW DO I LIST CDS.  
\*\*\*USE LIST.

WHAT PROGRAMX AVAILABLE?  
\*\*\*LIST, DUP, AND INTER.

WHAT DOS DUP DO?  
\*\*\*DUPLICATES CARDS.

HOW DO I DUPLIMATE CRDS.?  
\*\*\*USE DUP.

HOW DO I INTERPRETARE CRDS.?  
\*\*\*USE INTER.

WHAT DO INTER DO  
\*\*\*INTERPRET CARDS.

HOW DO I LST CRDS?  
\*\*\*USE LIST.

WHAT DOES LIST DO????  
\*\*\*LISTS CARDS.

WHAT DEOS INTRE DO?  
\*\*\*INTERPRET CARDS.

HOW DO I SIGN OFF.  
\*\*\*HIT CARRIAGE RETURN!!

At the heart of the adaptive query program is an algorithm for evaluating the similarity of two character strings. This algorithm was developed by G. L. Rouse, D. C. Gause and the author. An application for a patent has been made. A description of the algorithm is now given. Call the character strings to be compared A and B. Then:

Step 1:

Form a matrix, M, by assigning a 1 to M[I;J] if and only if A[I] = B[J]. Otherwise M[I;J] is set to 0; This matrix is formed by the following APL expression:  $M \leftarrow A \circ . = B$ . For example, if A 'ANNE' and B 'ANNIE' then M is:

```

      A N N E
A    1 0 0 0
N    0 1 1 0
N    0 1 1 0
I    0 0 0 0
E    0 0 0 1

```

Step 2:

If a row or a column of M contains more than one 1, then retain only the one closest to the main diagonal; the following APL expression does this:

$M + N = \Phi((\rho B), \rho A) \rho (x / \rho M) \rho SS + 0 = SS + \lceil / (N + M \times 1000 - |(\rho A) \circ . - \rho B)$

Note that if two 1's are equidistant from the diagonal, the expression would retain both.

From the preceding example we would get:

```

M = 1 0 0 0
    0 1 0 0
    0 0 1 0
    0 0 0 0
    0 0 0 1

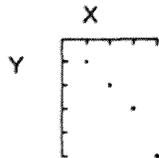
```

Step 3:

Consider the 1's in M as points on an X-Y coordinate system. That is, if M[I;J] is equal to 1 then we have a point with the Y-coordinate equal to I, and the X-coordinate equal to J. The APL expression for this is:

$X + D / (S + \rho D +, M) \rho \rho B$   
 $Y + D / \Phi((\rho B), \rho A) \rho S \rho \rho A$

From the preceding example we would get:



Step 4:

The standard correlation coefficient (which measures linear dependence) of the points is taken as a measure of similarity between the two strings. The closer to 1, the greater the similarity, the closer to -1 the greater the difference. The following APL expression evaluates the standard correlation coefficient:

$CC + ((N \times + / X \times Y) - X1 \times Y1) \div (((N \times + / Y * 2) - (Y1 + + / Y) * 2) * 0.5) \times ((N \times + / X * 2) - (X1 + + / X) * 2) * 0.5$

For example, consider the results of applying this algorithm to determine the similarity of the question 'WHAT IS TODAY?' with several other phrases:

Phrase	Correlation Value
WHAT IS TODAY	1.00
WHTA IS TODAY	0.994
WHAT IS TODAY	0.997
WHAT IS TDAY?	0.997
WHAT TODAY	0.97
WHAT DAY	0.92
TODAY IS WHAT	-0.05
YATOD SI TAHW	-0.36
MY NAME IS ED	0.001

With this algorithm in hand, the implementation of the Query System is fairly straightforward. A table is kept of questions seen and their associated answers. Associated with each question is a threshold which the correlation value must exceed in order for verification not to be requested. This verification threshold is adjusted so that verification is not requested more than once for any given input question. Also associated with each question is a threshold value which the correlation value must exceed for the question to be considered a match. When a new question is entered, a correlation value between that question, and all the questions in the table is computed. Only questions whose correlation value is higher than the associated threshold value are considered as candidates. Amongst the candidates, the one with the highest correlation value is chosen, and the answer associated with that question is given. If the answer given turns out to be incorrect, the threshold value associated with the selected question is raised to be slightly higher than the correlation value obtained for that question, thus insuring that the question would not be a candidate when the same question is posed to the system. Also, the new question is stored and the program asks what the correct answer to it should be, thus, another entry into the question-answer table is made. The threshold initially associated with a new question is set to a "low value."

The attached flowchart gives a more detailed description of the program.

Note that this adaptive query system has many applications; some possible uses include:

- allowing CAI (computer aided instruction) users to ask questions, at any point, about the subject being taught,
- questioning a system (such as APL) to find the type and use of available commands, and
- ignoring simple spelling errors in compilers.

