Diagnosis can help in intelligent tutoring

Roderick I. Nicolson
Department of Psychology
University of Sheffield
Sheffield S10 2TN, England
email: R.NICOLSON@UK.AC.SHEFFIELD.PRIMEA

Abstract

Recently there has been controversy about whether Intelligent Tutoring Systems are, even potentially, more effective than standard CAL programs, that is, whether it is educationally more valuable to attempt to identify the cause of user's mistakes rather than merely explain the correct method. This issue was addressed by comparative testing of two versions of the SUMIT Intelligent Tutoring Assistant for arithmetic using a diagnostic version, which diagnosed errors and gave appropriate messages, and a 'CAL' version was identical in all respects except that it made no diagnoses and therefore gave standard error messages indicating the correct method. In a comparative study of the two versions, a class of 9 year old children were first divided into two matched groups on the basis of a pencil and paper pre-test, then both groups had two 30 minute individual sessions with the appropriate version of SUMIT, and then performance was assessed on a subsequent pencil and paper post-test. Both groups improved significantly in their performance from pre-test to post-test, but the diagnostic group showed significantly greater reductions in the number of bugs. It is concluded that diagnostic remediation can be more effective than non-diagnostic approaches.

Introduction

Traditional Computer Aided Learning (CAL) programs have been criticised on the grounds that they do not *understand* the domain for which they were devised, and so they cannot give the adaptive help expected of a human teacher. This critique proved the stimulus for the creation of Intelligent Tutoring Systems (ITSs) which did understand their domain sufficiently to provide the same adaptive quality of guidance and instruction as a human teacher. Intelligent Tutoring Systems have made impressive progress in the intervening years, making contributions not only to pedagogical theory but also allowing empirical tests of theories of learning (see Anderson et al., 1990, for a recent review). However, the educational credibility of the ITS approach has recently been called into question by Sleeman et al. (1989), who were evaluating the effectiveness of remediation by human tutors in the domain of linear algebra problems. In a series of studies Sleeman and his colleagues compared the effectiveness of 'model-based remediation' (in which the tutor identified the type of error made, and explained why it was wrong), with 'reteaching' in which the tutor ignored the type of error made and merely explained the correct procedure. Both procedures were effective (as compared with a control group who received no remediation), but they were equally effective, leading the researchers to conclude that *"when initial instruction and remediation are primarily rule-based and procedural, remedial reteaching appears to be as effective as model-based remediation. From this it follows that 'classical' CAI would be as effective as an ITS"*. (1989, p563).

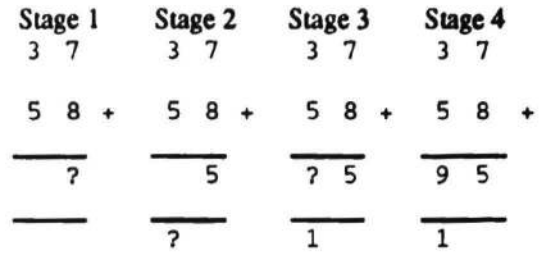
Recently I have developed the SUMIT system which is intended to function as an 'Intelligent Tutoring Assistant' for early school arithmetic (see Nicolson, 1990 for a full description of the design issues and studies of its effectiveness). SUMIT provides an ideal opportunity to assess the added value of diagnosis in tutoring in that diagnostic feedback is normally available, but can be 'turned off' if required by setting the appropriate flag. Both versions are otherwise identical, with the non-diagnostic version (henceforth SUMIT-ND) giving support in terms of the correct way to answer the problem, and the diagnostic version (henceforth SUMIT-D) giving not only that support but also a brief diagnosis of why the user's answer was wrong. The design of the study is therefore straightforward. We took a class of 9 year old children, gave them a pencil and paper pre-test on subtraction sums selected to investigate a range of potential problems, ranked them in order of score, split the class into two matched groups via this ranking, gave group 1 two sessions of individual practice with SUMIT-D, and group 2 two sessions with SUMIT-ND, then gave them a pencil and paper post-test equivalent to the pre-test, and compared the resulting gains in score and understanding. Before describing the study in detail, it is valuable to provide some more information on the SUMIT system.

The SUMIT Intelligent Tutoring Assistant

SUMIT was inspired by Brown and Burton's seminal work (1978) on diagnosis of the reasons underlying arithmetic errors, which led to the creation of the DEBUGGY system for bug diagnosis. In many ways, their research program was exemplary cognitive science, starting with identification of an important theoretical issue, collecting a large corpus of human performance data relating to that issue (children's substraction errors in this case), then constructing an offline diagnostic system intended to infer from the errors manifested which procedures were not fully understood, thus moving from performance assessment to competence assessment. The approach proved very fruitful, to the extent that most subsequent ITSs incorporated a 'bug catalogue' as part of their diagnostic armoury, and also in providing a rich source of data and ideas for important theoretical developments such as VanLehn's Sierra theory of procedural learning (e.g., 1990). But noone actually constructed a working, fully interactive, ITS for school arithmetic! SUMIT was the result of a longstanding 'spare time' project, conducted jointly with Margaret Nicolson, an experienced teacher of middle school arithmetic, to do just that.

The development program followed an 'evolutionary' strategy. Extensive knowledge engineering studies were undertaken over a period spanning three years in which first a detailed analysis of the traditional methods of teaching arithmetic was performed (based on three classroom studies). These studies were intended to identify areas of strength and weakness in the traditional approach, thus allowing the program to be targetted on relief of the weaknesses of traditional teaching, rather than duplication of the strengths. In particular, we identified the ability to give immediate feedback as critical, together with the ability to generate sums at a difficulty level appropriate for the child. These two capabilities would essentially allow a child to get on with practice at sums without the need for continual checking by the teacher. By contrast, the ability to explain why the methods used were the appropriate ones seemed much better suited to the traditional classroom demonstrations, where the teacher was able to use a range of techniques, adapted to his/her preferred teaching style and to the capabilites of the children, to explain the basis of the procedures. The analysis led us to undertake the construction of an 'Intelligent Tutoring Assistant' (ITA), less ambitious than an ITS, aimed at providing adaptive, generative practice at the procedural skills, with support for *which* procedure to use, and *how* to do it, but not for *why* the procedure should be used. The ITA was aimed to assist, rather than replace, the teacher.

Figure 1. The traditional stages in pencil and paper addition



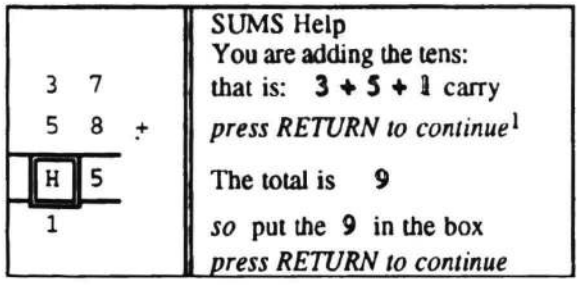
The analyses also identified the target skills required, a teaching strategy for imparting them, and a range of teacher preferences important for smooth incorporation of the ITA within the traditional teaching methods. A non-diagnostic program (SUMS) was then developed and tested extensively in the school setting. This investigation led to the introduction of further teacher support facilities, but its main function was to collect automatically a large corpus of arithmetic mistakes made in free use of the program for each of the four operations — addition, subtraction, multiplication and division. Extensive hand analyses were carried out on the corpora, leading to the identification of the error types (including their incidence), and, following analysis of how to automatically diagnose the major bugs, we were then able to 'bolt on' an online diagnostic capability, thus creating the SUMIT prototypes. Further details are provided in Nicolson (1990).

Using the SUMIT system

The following description shows how SUMIT is able to give a reasonably faithful replication of the traditional approach to arithmetic. Figure 1 demonstrates the traditional stages in pencil and paper arithmetic. The sum is written down on paper and the computation is carried out in stages as shown below — from units through carries to tens. The question mark is, of course, imaginary and it is included here to indicate which stage is involved. A clear difference between this written arithmetic and mental arithmetic is that it occurs step by step and, most important, intermediate steps are explicitly entered. Completion of the sum is often accompanied by muttered self-instructions somewhat like seven add eight is ... fifteen, so write down the 5 (stage 1) and carry the 1 to the tens (stage 2). Now three add five is ... eight, add the one carried, that's nine, so put the 9 in the tens (stage 3)'. Exactly the same procedure is used by SUMIT, with the child required to complete all five stages in the appropriate order, and if no mistakes are made, the procedure is essentially identical. Following successful completion of a sum, SUMIT generates a further sum at the appropriate difficulty

Figure 2. The non-diagnostic adaptive help available in the SUMS program

(The user has nearly completed the sum, but is unsure how to complete the addition of the tens column, and so presses H. The right hand side illustrates the help given in such a situation.)



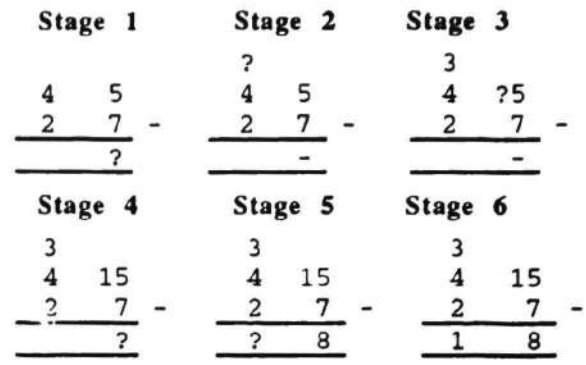
level and so on. The advantage of CAL becomes apparent if a mistake is made. Since the appropriate answer is always known for each stage in completing the sum, any error is noted immediately, and the user is warned of the error and required to try again. In the original SUMS program, adaptive help was available either on demand or following three errors on a given sum, but this only explained the correct method for continuing the sum, and made no effort to diagnose what the user's misconception might have been. On the basis of the extended studies of performance on SUMS, SUMIT is able to diagnose up to 20 different bugs for each of the four arithmetic operations. This allows an immediate diagnosis of the likely cause of any error. For instance, if the user typed in '8' instead of '9' at stage 3, the program decides that the most likely bug is 'failure to add in carry' and is therefore able to offer the suggestion 'Remember to add in the carry 1'. Adaptive standard help is again available on demand or after three helps (see figure 2).

In both diagnostic and non-diagnostic versions, an error results in a warning tone, and the user is not allowed to proceed until the correct answer has been entered. Non-diagnostic help is normally given automatically following three errors on a sum. Diagnostic help following an error of typing in '8' in the above situation would involve the short message "Remember to add in the 1 you've carried from the units". In the non-diagnostic form, following an error only the warning tone is presented, followed by the message "Bad luck, please try again".

In view of the greater complexity of subtraction, and in recognition of its special status in the ITS literature, for the investigation of diagnostic versus standard feedback we decided to investigate the effects of diagnostic support on subtraction skills.

[1]Initially only the first part of the message is displayed. Pressing the Return key adds in the next part, and so on.

Figure 3. A subtraction sum which involves 'borrowing'



The standard 'decomposition' approach to teaching subtraction is shown in Fig. 3. Note the complexity of the procedures involved.

The self-instructions for this sum might go as follows: *"5 take 7 won't go, so put a dash in the box (stage 1) and try to take help from the tens column (stage 2). 4 take 1 leaves 3, so cross out the 4 and put 3 (stage 3). Next take the ten help we were given [borrowed] and give it to the units — that makes 15 (stage 4). We can now take the 7 from our 15, that makes 8, so put the 8 in the box (stage 5) and go to the tens column. 3 take 2 is 1, so put the 1 in the box (stage 6)".*[2]

The commonest subtraction bug (1S) occurred at stage 2, where rather than subtracting 1 from the tens, the child got confused and performed the subtraction on the tens column (thereby yielding 2 in this case). Bug 1 appears to occur only on the computer, and the most generic subtraction bug (2S) [smaller from larger] occurs at stage 1, where the child enters *2* for *5-7*. This is a beginner's error, symptomatic of difficulty in knowing how to cope with a negative outcome.

The bugs diagnosed by SUMIT-D, and their incidence in the initial corpus are shown in Table 1. Note that the use of —> in the example indicates that the user entered the digits in the order shown. For instance, for bug 1S, the sequence for answering 83-24 was - {correct}, then 6, {the error, reflecting subtraction of the two entries in the tens column (8-2) rather than subtraction of the borrowed 1 from the tens column, leading to the answer of 7}. It is much easier to follow this exposition if the sum is laid out as shown in figure 1!

[2]The 'decomposition' procedure for subtraction is now preferred to the older 'equal additions' method which would add 10 to both top and bottom (ie turning the 5 of 45 into 15, and turning the 2 of 27 into 3) on the grounds that for decomposition the manipulation is only on one number, and can easily be shown to be valid by means of Dienes' blocks etc.

Table 1. Subtraction Bugs Diagnosed by SUMIT-D

Bug	Description	Example
1S	Subtract current column in mid-borrow	83-24 —> -6 etc.
2S	smaller from larger	3-6=3
3S	Put 1 in before decrementing column	83-24 -> -1 etc.
4S	miss out stage in initial borrow	83-24 -> 7 etc.
5S	Don't decompose 10 in borrowing over 0	803-24 —> --71x etc where x<>9
6S	0-n=n (specialis'n of 2S)	0-7=7
7S	0-n=0	0-7=0
8S	use non-decremented minuend	583-124—>-7196
9S	Lose place in mid-borrow	83-24 —> -79
10S	'Add 10' bug	83-24 —> -710
11S	response perseveration (repeat prev press)	803-24—>- - -
12S	subtract 1 'for luck' from last column	83-22=51
13S	Missed out step	eg. 10-7=3
14S	0-1=0 when borrowing across 0	803-24 —> -0
AS	arithmetic error	13-6=8
US	*unclassified (non-borrow)*	
UbS	*unclassified (in borrow)*	

Experiment. Diagnostic help vs non-diagnostic help using SUMIT

Two groups of 9 year old schoolchildren from the same class were selected, individually matched on performance on a pre-test. Both groups then experienced two 30 minute individual sessions of SUMIT, one group using SUMIT-D and the other group using SUMIT_ND (with the standard feedback and help facility). Children used the program individually, with two children at a time taken out of their normal arithmetic lesson. The experimenter was Chris Harrop, a third year undergraduate student, who had chosen to undertake the work as part of his final year undergraduate dissertation in Psychology. The experimenter's role was to ensure that the appropriate version of SUMIT was selected, to check that each child started the session at an appropriate level of difficulty, and to provide general encouragement. He gave no direct instructional support. In the first computer session each child started at the baseline level, and sums were automatically generated at levels of increasing difficulty until mistakes started to emerge, at which stage the program generated sums of the appropriate difficulty subsequently. In the second session the child was encouraged to start at a level one below that reached in the first session. Finally, performance on a pencil and paper post-test equivalent to the pre-test was measured. For each test the written answers were scored, and any error made was assigned to one of the bug categories (see Table 1). The total bug count was determined by including the unclassified bugs (which correspond to a bug not included in the diagnostic help) but not the arithmetic bugs. Comparison of pre-test and post-test scores and bugs for the two groups should reveal whether diagnostic help really does help or not..

Results

Results for the pre-test and post-test scores are shown in Figure 4a and those for bugs in Figure 4b. It may be seen that, as expected, both groups improved as a result of the sessions with SUMIT, and that the diagnostic group improved somewhat more in overall score, and markedly more in terms of the overall bugs. An analysis of variance on the scores indicated a significant main effect of time-of-test ($p<.01$) but no significant main effect of group, and no significant interaction. In terms of the effectiveness of the learning induced, the non-diagnostic group's mean score improvement was 0.30 sd units [based on the original standard deviation of scores of both groups together, cf. Bloom (1984)], well below that of the diagnostic group (0.75). An analysis of variance on the bugs data (omitting children who obtained pre-test scores of 29 or 30 out of 30) indicated a significant main effect of time-of-test ($p<.05$), no significant effect of group, but a significant interaction between group and time-of-test ($p<.05$), indicating that the diagnostic group eliminated their bugs significantly more effectively than the standard group. The individual results are displayed in Figure 5. Comparing the histograms for the two groups, it is clear that the major effects are attributable to those children who were initially performing badly. For the diagnostic group, there are large improvements (see especially OT who improved from 5/30 to 30/30), whereas this improvement was less consistent for the non-diagnostic group.

Discussion

It remains to consider the wider significance of these results. First it is important to stress that the results relate only to two groups of children in one school on one task, and that the results are attributable to only a few of these children. Next, the major improvement is attributable to the SUMIT program itself, and the further improvement due to the diagnostic element is of only secondary importance.

Fig. 4a. Scores for the two groups

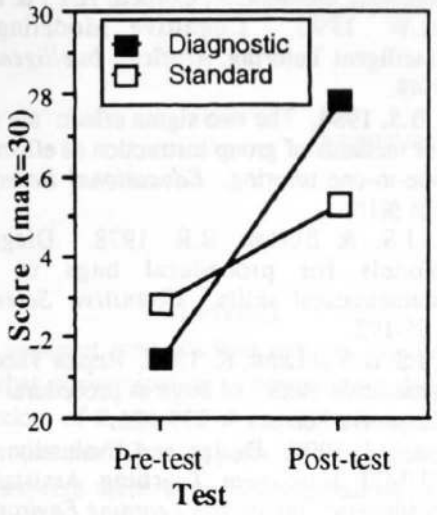


Fig. 4b. Bugs for the two groups

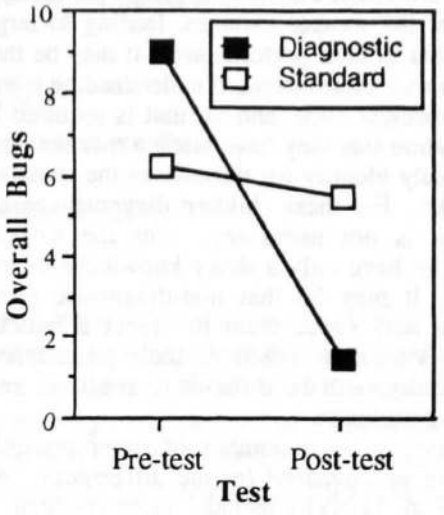
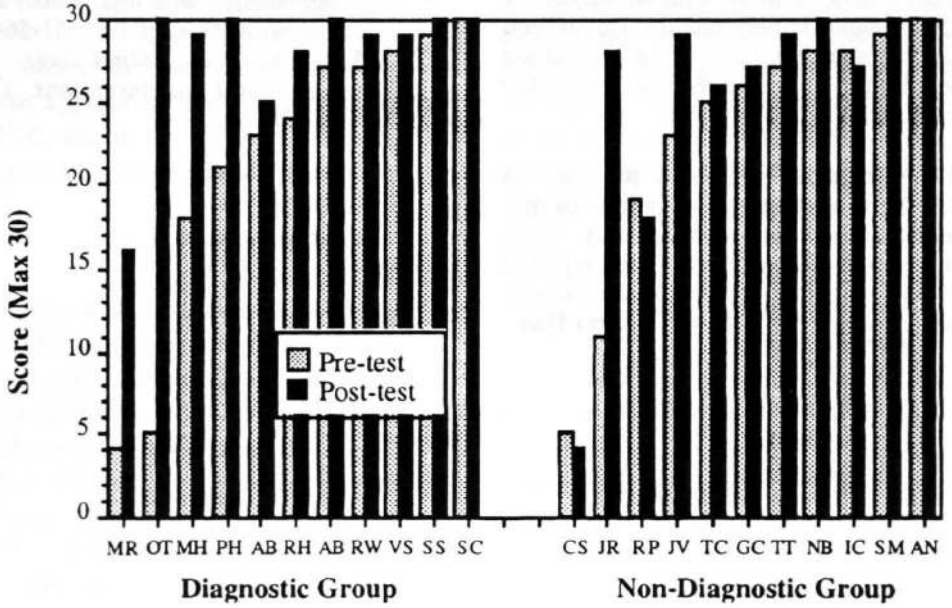


Figure 5. Individual Scores at pre-test and post-test



Further studies are needed to assess the reliability and the generality of these results. If the results are representative, one must consider why this study obtained a differential effect, unlike the three studies on highschool algebra reported by Sleeman et al. (1989):

(i) our differential test was more sensitive, in that diagnosis was the only factor differing between the two conditions, whereas for Sleeman et al. human tutoring was involved, which may have increased the variability of the effects

(ii) the arithmetic diagnosis was explicitly linked to corpora of data collected in previous studies, and thus likely to be well-tuned to the types of mistakes made.

(iii) the arithmetic diagnostic messages were very brief and to the point, whereas the 'model based remediation' used in the algebra studies was a lengthy

process. Clearly, the latter would reduce the time available for 'reteaching'.

(iv) the arithmetic diagnosis appeared particularly valuable for the weaker children, leading to large improvements in their performance. It may be that for children with more advanced understanding many errors are careless slips, and all that is required is some indication that they have made a mistake, and they can easily identify for themselves the cause of their mistake. For these children diagnostic-based remediation is not necessary. For the weaker children, who have only a shaky knowledge of the procedures, it may be that non-diagnostic error information may cause them to invent a 'patch' (Brown & VanLehn, 1980) to their procedures, which, if faulty, will be difficult to eradicate and cause lasting confusion.

Of course, a large number of other possible reasons may be advanced for the differences. As Sleeman et al. (1989) conclude, more research is needed to identify those situations in which diagnosis-based teaching is more effective. We conclude that although SUMIT is effective in helping children learn the rules of arithmetic with or without diagnostic help, SUMIT's diagnostic help facility does indeed confer a further advantage in terms of the elimination of bugs, especially for those children who are weaker at arithmetic.

Acknowledgments. It is a pleasure to acknowledge the contributions of many of my students over the years, and, most recently, Chris Harrop, who carried out the empirical work reported here. My thanks also to Lydgate Middle School, Sheffield, and in particular to its head, Geoffrey Hall.

References

Anderson, J.R.; Boyle, C.F; Corbett, A.T., & Lewis, M.W. 1990. Cognitive Modeling and Intelligent Tutoring. *Artificial Intelligence* 42: 7-49.

Bloom, B.S. 1984. The two sigma effect: the search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher* 13: 3-16.

Brown, J.S. & Burton, R.R. 1978. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science* 2: 155-192.

Brown, J.S. & VanLehn, K. 1980. Repair Theory: A generative theory of bugs in procedural skills. *Cognitive Science* 4: 379-426.

Nicolson, R.I. 1990. Design and Evaluation of the SUMIT Intelligent Teaching Assistant for Arithmetic. *Interactive Learning Environments* 1: 265-287.

Sleeman, D.; Kelly, A.E.; Martinak, R.; Ward, R.D., & Moore, J.L. 1989. Studies of diagnosis and remediation with high school algebra students. *Cognitive Science* 13: 551-568.

VanLehn, K. 1990. *Mind Bugs: the origins of procedural misconceptions*. Cambridge, MA: MIT Press.