Promoting productive language learning strategies in an implementation of computer-based cloze through an investigation of on-line ESL learner interaction

Draft Chapter: Data Collection and Analysis

by Vance Stevens, Sultan Qaboos University

The following constitutes a report of progress in data collection and analysis, submitted via Graham Davies (Supervisor) to the Research Degrees Committee, School of English Language Teaching, Thames Valley University, submitted **November, 1993.**

Almost all of the 500 freshman students admitted to Sultan Qaboos University (SQU) each year take courses at the Language Centre (LC). During class time, students often use the Student Resource Centre (SRC), where they have access to, among other things, computers to help them improve their English. The present researcher, a lecturer in the LC in charge of managing the SRC, sees to it that CALL materials are maintained there appropriate to these students.

SuperCloze was developed at Sultan Qaboos University (SQU) by Vance Stevens and Steve Millmore as one of several text manipulation templates capable of delivering a seemingly infinite number of language learning exercises from texts provided by coordinators of LC courses. SuperCloze has been running in data collection mode in the SRC for several years now. During that time, the program has undergone numerous revisions, and the data collection component has been made increasingly more sophisticated.

Students use SuperCloze in the SRC as part of a class group, in which case they might be specifically instructed by a teacher to use the program, or during evening self-access sessions, at which time students might use the program on their own volition. In either case, the program has always appeared popular as a CALL activity with both students and teachers. Perhaps because cloze is an accepted classroom activity, teachers understand and agree with the underlying principle and tend to recommend its use to their students. Although no records are kept in the SRC of which other programs are used there, observation suggests that students themselves often select SuperCloze in preference to the other text manipulation programs available.

When taking students to the SRC as a group, teachers rarely monitor closely what they do on the computers there. Students are never asked to identify themselves at the computers, and they are unaware that the program is monitoring their keypresses (which in any case are not traceable to individuals). Therefore, the students in the SRC approach a very high degree of self-access use of SuperCloze, especially in the self-access sessions where they themselves choose what programs to use. Consequently, we have in the SRC at SQU an opportunity to study how a

text manipulation program might be used if purchased shrink-wrapped and taken home by a language learner. In other words, we stand to learn how such programs are used when users feel no need to accommodate an authority figure such as a teacher or researcher, but use the program purely for their own benefit or, as sometimes appears the case, as a means of making something happen on a screen, as if watching TV.

The purpose of the present study is to determine how students use the SuperCloze program in the SRC, and how the program can be adjusted to optimize that behavior for language learning. The evidence at hand is a record of student keypresses written to disk each time the students use the program. This chapter describes how this evidence is gathered and analyzed.

DATA COLLECTION

SuperCloze evolved from a cloze component in Text Tanglers, a battery of text manipulation programs on which Steve Millmore and I began working in 1986. By 1989, we had a version of SuperCloze which would write data to disk. Prior to that, we had developed and refined our technical skills on a data-collecting version of HangMan, another module which started out in the original Text Tanglers set. Analysis of the data from HangMan suggested numerous improvements to the program and led to the development of a stand-alone module which we call HangMan-in-Context. HMIC emphasizes productive strategies in HangMan such as starting solution of a problem with vowels or with a single hint. Also, it discourages strategies disruptive to language learning; for example, it limits the use of hints and penalizes thoughtless keypresses both in the point system and by display of an on-screen graphic. As far as the present project is concerned, work on HangMan has exercised a pattern of development that might be applied to SuperCloze: (1) implementation of a prototype CALL program, (2) data collection and analysis during trial on students, and (3) development of an improved version of the program which can be shown to be more pedagogically sound than the original.

Despite some similarities, data analysis with SuperCloze has proven much more problematic than with HangMan. Keystrokes in HangMan were relatively straightforward to quantify, and whereas I was able to write a post-processor accompanying HMIC to yield automatic analysis, with SuperCloze, it is much less clear what there is to count in the data, and having determined that, how to tell the machine where the data are in the data file and how to interpret them. So far, I have not been able to write a program that can get at what I am looking for as well as I can manually; that is, the frustration levels reached over the time it takes to program and debug the kind of post-processor I need have been so far greater than those endured painstakingly picking apart the data by hand. However, I do hope with more practice (at both endeavors) to produce a post-processor that will replicate my findings and then be able to do a speedy analysis on the masses of data that can be collected when the program is simply left running somewhere.

One complication is the structure of the data files accumulated by the SuperCloze program. As more is learned about collecting and analyzing this kind of data, improvements are made to the data-collection module of the program that in turn require revisions to the latest

version of the post-processor. There have been several such improvements in recent versions of the program. For example, the first version of SuperCloze that collected data (Version 1.9) printed all the words in the original passage in a column down the file. Although all the elements were there, like pieces to a puzzle, it was difficult to reconstruct the original passage as it must have looked to the student (reconstruction by hand was performed to produce the report I made at TESOL on this topic). So one of the first improvements to the data collection facility was to get the program to display the cloze passage in the data file in its puzzle form.

Keeping Time

The first problem with data analysis is to discern individual student sessions, that is, which chunks of data represent the work of one or more students interacting with Super Cloze in one sitting. The first versions of the program did not include a time stamp as part of the data, and in the resulting data it is impossible to know with any reasonable degree of accuracy when a student has quit and resumed work or quit and left for the day. In the first data-collecting versions of SuperCloze, it was assumed that when "Quit" was encountered in the data, followed by "Logon", this denoted a new student session. However, observation of students at work suggested that they were frequently liable to quit and then log themselves back on, or turn the machine completely off and then back on again and resume working.

Whether a new or different student was responsible for the "logon" could be discerned if a record were kept of the time, so a new version of the program was created that printed time$ stamps to the file. The first version of this new system yielded notations in the data that, for example, the student interaction was "Correct at 00:00:00" or "See Solution pressed at 00:00:00." Now the information was there in the file, but the question was where to find it. The answer was to put the time stamps at column one, where not only could it be easily found by the post-processor, but the next item of useful information on any line would be predictably ten bytes over. It seems obvious in retrospect that things should have been set up this way from the outset, but it is to gain the advantage of hindsight that such projects are piloted. (Consequently, all data collected in the pilot years of this project have been set aside in favor of more recently collected data, which gives time-stamp clues as to which chunks of data are likely to be the work of the same student or students. There are plenty of data, both new and old; the problem is only to analyze it.)

A final development in the collection of useful data has been the recent installation in the SRC of ten Leo 386 computers with systems clocks that keep time. The XT computers on which most of the data had previously been collected lack clocks, and though they keep time in the manner of a stopwatch, they don't give theactual time or date. With the new computers, it has been possible to interpret whether recorded sessions took place on the same day or not, and also to get some idea of the frequency of use of the programs. I find for example, from my most recent batch of data, that use of SuperCloze on a given computer in the SRC can be as infrequent as once a month, and that maybe for a few minutes or seconds! Before computers with date stamps wereavailable, I had assumed that SuperCloze was used much more frequently than that.

In this last sample of data (the first to be gathered off the new computers), I also find that the clocks on the new computers are not accurate, but if they were, then it would be possible to make better guesses about conditions under which students were using the programs. Given the date and time, we could track the kind of student (medical or remedial, for example) whose class was supposed to be using the computers at that time, and we would know if the session took place in a class or evening self-access setting. Accurate time-keeping would also help explain instances in the data where SuperCloze interaction stops only to resume five or ten minutes later (same date). It would be useful to know the exact time - if the lag time occurred during the day and during a break between classes, then the resumption would be the work of a second student; if during the middle of a class hour, it would more likely be the case that the student did something else on the computer and then came back to SuperCloze. If the break were during evening self-access, it might suggest that the same student was browsing elsewhere on the computer and then returned to SuperCloze, or it could mean that another student had come along and found a computer running, and then decided to see what had been left there for him to do, thereby reactivating time-stamping.

Whatever the assumption, it is clear that as manager of the SRC, I should take care to see that the systems clocks on the ten new computers are kept up to date. Before the latest analysis of the data, this had not occurred to me. The computers in the SRC are at present not networked, but if they were, then accurate time could be kept by simply monitoring the time on the server.

Network possibilities, and considerations

I have just installed SuperCloze and the other text manipulation programs developed at SQU on a networked system in the newly opened College of Commerce and Economics at Sultan Qaboos University, which has about 100 computers available to the 120 business students throughout the new building (this number of students is expected to increase dramatically in coming years). The installation includes HangWord, a whole-text deletion program recently developed by Steve Millmore and I, which does not at present collect data, but which could be configured to do so (theoretically, time permitting). Once this is done, we will have further insights into strategies employed by students in this genre of CALL.

However, a networked system poses problems in data collection (as the data are presently collected to a single file) which could no doubt be overcome, but which I have up to now been spared having to deal with in collecting the present batch of data. As it would be interesting to monitor use of the text manipulation programs in the network environment, there is some incentive to configure them to do so. At the moment, no data are being collected on the network.

DATA ANALYSIS

Work with the data generated by the program suggests two areas of analysis. One lends itself to number crunching, and it is this aspect of the data which I am focusing on at present, as I

would like to produce this elusive post-processor which might wring from the files masses of insightful information. This aspect of the study deals with how much of the available text students appear to be working on, whether they approach the text linearly or holistically, how much time they spend with the text, how they use the help features provided in the program, and so on.

Another area of analysis concerns the mistakes students make and how they deal with feedback in the program. This line of inquiry regards distinctly human properties of language and so defies the kind of definition that one could build into a computer. One interesting aspect of the present research is that correct answers given by students are among the least revealing of the data collected. In this study, only mistakes and requests for help provide clues as to the strategies students use in arriving at correct answers.

At a later date I would like to pursue more closely this second line of inquiry. However, the present analysis explores what students do in general with the SuperCloze program. Toward this end, I have taken a sample of 100 cloze paragraphs chosen virtually at random from among the thousands I have data on. That is, I selected three files from two or three computers in the SRC, one of which happened to be from Version 3.3 of the program, and the other two from Version 4.1. I then went at the data in those files until I had elucidated 100 paragraphs' worth of interaction. The number 100 was chosen because it seems a robust representation of the data, and for convenience in calculating percentages. The kind of analysis arrived at might in future be done more speedily on computer, but I find it has to be done first by hand in order to develop a feel for the data and find in it what items will be worth counting.

Presentation: What to count ... ?

The analysis is summarized in an accompanying table. The leftmost column (Par#) numbers the 100 paragraphs that were analyzed, and the next column (Code#) is my reference to where these paragraphs occur in my data. Several things can be seen from the code numbers: (1) "41" and "33" give the versions of the program that produced the data (these versions were fairly similar as far as our students were concerned). (2) The next numbers denote a "session" recorded; that is, they group the work of one of more students who kept the computer running over a continuous period of time - and (3) where a, b, c, etc. appear at the end of a number, this distinguishes the different paragraphs, or cloze passages, worked during the same session.

The next column over gives the number of sentences that appear to have been considered by the student in working that particular cloze passage. In other words, if a student spent just seconds solving only the first gap in the paragraph before quitting, then he or she probably did not look beyond the first sentence in the paragraph. Thus, the number here tells in what sentences the gaps solved by the students appeared. Notations in this column are generally "1" (the student appears to have looked only at the first sentence), "all" (students considered all sentences, because all gaps were solved or attempted), or something like "2//7" (meaning the student addressed gaps found in two of the seven sentences in the passage). The word "global" here means the student attempted gaps at various places in the paragraph.

The column headed "seq" tells whether the student addressed the gaps linearly ("lin") or non-linearly ("n/l"). The next three columns give the number of gaps solved successfully ("ok"), the number of gaps attempted (often equal to or one more than the number solved), and the number of gaps in that particular passage.

The next column gives the time to the nearest tenths of a minute (6 seconds) that the student spent on that paragraph. As can be deduced from the table, 334 minutes of interaction with the program were examined.

The next columns quantify the help requested. Two kinds of help are possible (besides an actual Help screen, which only gives instructions for working the program). One kind of help is "See Solution", which allows students to view the non-degraded paragraph and then either return to the passage or skip it and go on to the next one. The numbers under "#ss" give the number of times students saw the solution while viewing that paragraph, and the next column (%s/ok) compares this value to the number of gaps solved correctly. A student with a 25% figure in the latter column would have used See Solution in a quarter of the successfully solved cloze gaps for that passage.

The other kind of help is "Hint", which reveals the letter at the cursor position. Under #h and #c are given the number of hints requested and number of characters in the target word, and the ratio of the two is given as a percent under "%hints". The latter value is the average of all the #h/#c calculations, including those which may run in columns out of sight to the right beyond the page margin. A low value here suggests that students made judicious use of hints, whereas a value over 50% suggests abuse of the hint facility for items in that paragraph.

At the bottom of the chart are average and median values for the various columns. Median values are useful where averages are skewed by anomalous interactions.

DISCUSSION

Dealing with Non-Fruitful Sessions

What strikes one in the data is the high incidence of non-fruitful, "just-looking" interactions. These can be spotted on the table wherever there are zero problems attempted, or where the amount of time spent on a passage amounts to only a few seconds. Zero problems attempted means that a student or students logged on to the program and then suddenly logged off, or logged off after some time without engaging in any significant interaction (e.g. Par# 51, in which a student chose, looked at, and quit from 8 passages in succession over ten minutes' time). In my data, there are 33 instances of zero-problems-attempted, fully a third of the interactions recorded.

Some of these might indicate that a student wanted to look the text over before attempting it, as with Pars. #29-31, which in fact represent a student's looking at 8 paragraphs one after

another via the See Solution/Next Passage option for over 6 minutes before finally requesting a single hint (Par# 32) just prior to logging off.

Par# 32 is representative of another example of non-fruitful interaction, an "0 for 1" result, where the student performed some action regarding a gapped item, but without success (e.g. attempted a problem and got it wrong, or requested a hint, then quit, or both). In my sample data, there are 16 such items.

In summary, of the 100 close passages (Par#'s) examined, about half (33 + 16 = 49) got essentially nowhere. It is a matter of interest at this point only that this happens with half the passages attempted, and one function of an automatic post processor would be to identify and tally such interactions. In future I shall limit my focus on such interactions to developing some way of discouraging them, and in further analysis concentrate solely on what goes on in more fruitful sessions.

Detecting Patterns in Fruitful Sessions

The more interesting question is therefore, what goes on in the half the sessions that do involve some interaction with the computer? One observation is that, in 15 of the 51 such sessions examined, interaction was constrained to within the first sentence. This produces an encouraging outcome, as it means that over a third of the interactions with the program (the remaining 36) involved interaction with all or with a substantial portion of the passage presented. In fact, in 22 of the sessions recorded, students completed all of the blanks presented.

Another interesting observation is that of all cloze passages in the sample in which more than one gap was addressed, only six were addressed in anything but a strictly linear, solve-one-gap, go-on-to-the-next manner (and in at least one of those six, interaction was essentially linear, with the student backing up only once to fill in a gap skipped). This method of approaching CALL-based cloze has been noted in the literature, and I have already in fact been trying to think of some means of encouraging a more holistic approach to the exercise (any suggestions?).

Use of help is another aspect of the program on which the data cast some light. Although adequate help is deemed necessary in order to ensure that students are never stuck in the program, use of help as a crutch should be discouraged. An ideal use of this program's present help features would be a frequent but judicious use of Hints, with See Solution utilized in only the direst of straits.

In general, the data suggest that use of hints by the subjects in this sample was indeed moderate, with only isolated incidents of heavy use (see for example, Par#'s 44, 45, 70, 75 and 76). It would be easy to build into the program some check on hints (as is done with HMIC, which turns off the hint feature after half the word has been elucidated). On the other hand, it appears to me that the hint feature was under-utilized by most students, especially by the two-thirds who had little or no interaction with the program. Therefore, another improvement to

the program will be to develop some means of suggesting hints to students when they seem about to give up. Furthermore, expanded hints are envisaged; for example, the ability to get a masked KWIC concordance on an unknown target item.

Use of See Solution is more widely abused. Twenty-five of the 36 passages in which there was significant interaction registered some use of See Solution, and some of this was exhorbitant (e.g. Par# 33, where SS was used 3.5 times per gap solved). See Solution appears to act in the manner of a drug; students try faithfully to solve gaps until they "discover" the feature, at which point its frequency of use increases. A signature strategy for at least two different subjects was to use Hint to expose a single letter in an unknown word, perhaps make an attempt at solving the problem, but failing that (or sometimes directly, without overt attempt at an answer) using See Solution to get the rest of the word. Still another pattern (see Par#'s 29, 30 and 36) was to look at the solution, return to the problem, and still fail to solve the gapped item. Clearly, in future versions of SuperCloze, use of See Solution will have to be curtailed in some way.

Instances of healthy use of hints and See Solution were not unknown in the data. Par# 13 for example is one in which the student solved all gaps, resorting occasionally to reasonable use of hints, and skipping and later cycling back through gaps he or she couldn't solve the first time around. See Solution was used only at the end of the session, to reveal the word "Mashona", the name of a tribal group in Kenya.

FURTHER WORK

There is much more that can be said about what the particular use of help features reveal about elucidation strategies in solving computer-based cloze, but I have not had time for more careful analysis at this point. In future stages of data analysis, I would like to look more closely at the anecdotal evidence in the data, which I hope will expose patterns suggesting richer inquiry.

Work so far has helped me mainly to make some sense of the welter of data collected and refine what it is I am looking for there. Peparation of the above analysis has helped me to more clearly visualize where those data are in the files I am collecting, and how I might get at them with a post-processor program. I think I might now turn my attention to a post-processor, as how one might be constructed is fresh in my mind at the moment, and as the existence of one will enable me to automate what I have done so far in the collection of future data and devote more energy to the more subjective kind of analysis mentioned above.

Also, as has been noted, certain changes to the program have been suggested in this analysis, and implementation of some of these changes should take place before more data are collected and analyzed.

Signed (Nov. 22, 1993):

Vance Stevens, Candidate: 

Graham Davies, Supervisor: _____

Data Analysis: SUPER CLOZE						Vance Stevens, Nov. 1993								
Par#	Code#	#snt	seq	#ok	#att'd	#gaps	time	#ss	%s/ok	%hints	#h	#c	#h	#c
1	41-112	1	lin	2	3	10	0.8							
2	41-113			0	0		1.5							
3	41-114	1	lin	2	3	12	2.3	1	50.0%					
4	41-115			0	0		3.0							
5	41-116	1	lin	1	2	9	18.1							
6	41-117a	1	n/l	0	4	21	4.8							
7	41-117b			0	0		2.0							
8	41-117c	1		0	1	20	1.9							
9	41-117d	1		0	1	19	1.1			20.0%	1	5		
10	41-118	1		0	1	22	3.3							
11	41-119			0	0		0.1							
12	41-120a	1	lin	1	2	7	9.1	1	100.0%					
13	41-120b	all	n/l	15	15	15	25.0	1	6.7%	35.3%	1	4	1	7
14	41-121a	all	lin	12	12	12	5.2	1	8.3%					
15	41-121b	all	lin	19	19	19	14.7	5	26.3%					
16	41-121c	all	lin	16	16	16	6.3	2	12.5%					
17	41-121d	3//6	lin	11	12	23	11.0	3	27.3%					
18	41-122			0	0		0.1							
19	41-123	1		0	1	11	0.6							
20	41-124a	1		0	1	7	0.5			20.0%	1	5		
21	41-124b	1		0	1	22	1.0							
22	41-124c	all	lin	6	9	22	1.6			12.5%	1	8		
23	41-125a	2	lin	1	1	3	0.9							
24	41-125b	1		0	1	22	1.5			11.1%	1	9		
25	41-126	all	n/l	10	12	12	6.4							
26	33-1a	1		0	1		1.1							
27	33-1b	1		0	1		1.1							
28	33-2			0	0		0.8							
29	33-3a:f			0	0		5.9	6	0.0%					
30	33-3g			0	0		0.3	1	0.0%					
31	33-3h			0	0		0.1							
32	33-3i	1		0	1	15	0.8			14.3%	1	7		
33	33-4a	2//7	lin	2	2	7	8.0	7	350.0%	25.0%	1	4		
34	33-4b			0	0		0.2							
35	33-4c			0	0		0.1							
36	33-4d	1		0	1	7	1.2	1	0.0%					
37	33-4e			0	0		0.3							
38	33-4f	all	lin	7	7	7	4.5							
39	33-5			0	1	5	1.3							
40	33-8			0	0		0.2							
41	41-1a	all	lin	14	14	14	5.9	3	21.4%					
42	41-1b	2//3	lin	8	8	14	3.3	4	50.0%					
43	41-2a	all	lin	10	10	10	1.3			20.0%	1	5		
44	41-2b	all	lin	27	27	27	5.8			59.8%	1	1	5	6
45	41-2c	all	lin	6	6	6	1.9			45.8%	2	4	4	8
46	41-3	1	lin	1	2	7	1.5							

47	41-4			0	0		0.1							
48	41-5			0	0		0.3							
49	41-6			0	0		1.2							
50	41-7	1	lin	4	5	21	2.0							
51	41-8			0	0		10.2							
52	41-9			0	0		0.3							
53	41-10			0	0		0.3							
54	41-11a	all	lin	16	16	16	30.4	5	31.3%	33.3%	1	3		
55	41-11b	4//14	lin	4	5	23	4.2	1	25.0%					
56	41-12a			0	0		1.0							
57	41-12b			0	0		1.3							
58	41-12c	1	lin	1	2	18	2.5							
59	41-13	1		0	1	7	0.7							
60	41-14a	1	lin	1	2	12	2.3	1	100.0%					
61	41-14b			0	0		0.1							
62	41-14c			0	0		0.1							
63	41-14d	4//7	lin	7	8	15	3.9	3	42.9%					
64	41-15	1		0	1	17	0.9							
65	41-16			0	0		0.7							
66	41-17			0	0		1.0							
67	41-18			0	0		2.5							
68	41-20	1	lin	0	2	8	1.2			36.1%	1	2	2	9
69	41-21a			0	0	90	0.3							
70	41-21b	1		1	1	13	0.8			100.0%	3	3		
71	41-22	glbl	n/l	5	6	25	1.7							
72	41-23			0	0		0.1							
73	41-24a	2//3	lin	4	5	15	1.8							
74	41-24b	1	lin	2	3	29	0.6							
75	41-25a	4//14	lin	6	6	32	13.1	1	16.7%	60.5%	4	4	4	8
76	41-25b	1	lin	4	4	11	4.2	3	75.0%	52.3%	2	8	9	11
77	41-25c	1		0	1	21	0.2			10.0%	1	10		
78	41-25d			0	0		0.1							
79	41-25e			0	0		0.1							
80	41-25f			0	0		0.1							
81	41-25g	glbl	n/l	0	2	17	1.7							
82	41-26a	all	lin	6	6	6	5.2	1	16.7%	33.3%	1	3		
83	41-26b	all	lin	16	16	16	6.0	1	6.3%					
84	41-26c	all	lin	2	2	2	0.6							
85	41-27a	1		1	1	20	1.6							
86	41-27b	all	lin	20	20	20	4.8	3	15.0%					
87	41-27c	all	lin	12	12	12	3.2	5	41.7%					
88	41-28			0	0		0.3							
89	41-29	all	lin	12	12	12				33.3%	3	9		
90	41-30			0	0		0.8							
91	41-31a	2//6	lin	2	3	12	5.4			37.5%	1	4	2	4
92	41-31j	glbl	n/l	1	3	8	1.6			33.3%	1	3		
93	41-31u	2//2	lin	3	4	7	2.5							
94	41-31v	1	lin	1	2	11	1.5							
95	41-31x	1		1	1	3	1.1							

96	41-32b	1	lin	4	4	11	2.0	1	25.0%	50.0%	1	2		
97	41-33a	3//5	lin	11	11	26	12.5	4	36.4%	16.7%	1	6		
98	41-33b	all	lin	7	7	7	7.9	4	57.1%	27.4%	1	10	1	7
99	41-33c	all	lin	12	12	12	6.8	5	41.7%	36.1%	1	2	1	3
100	41-39b	all	lin	8	8	8	6.7	4	50.0%	20.8%	1	3	1	6
Averages				3.45	3.92	15.242	3.34	2.8	44.0%	33.8%				
Median Values				0	1	12.5	1.5	3	26.8%	33.3%				