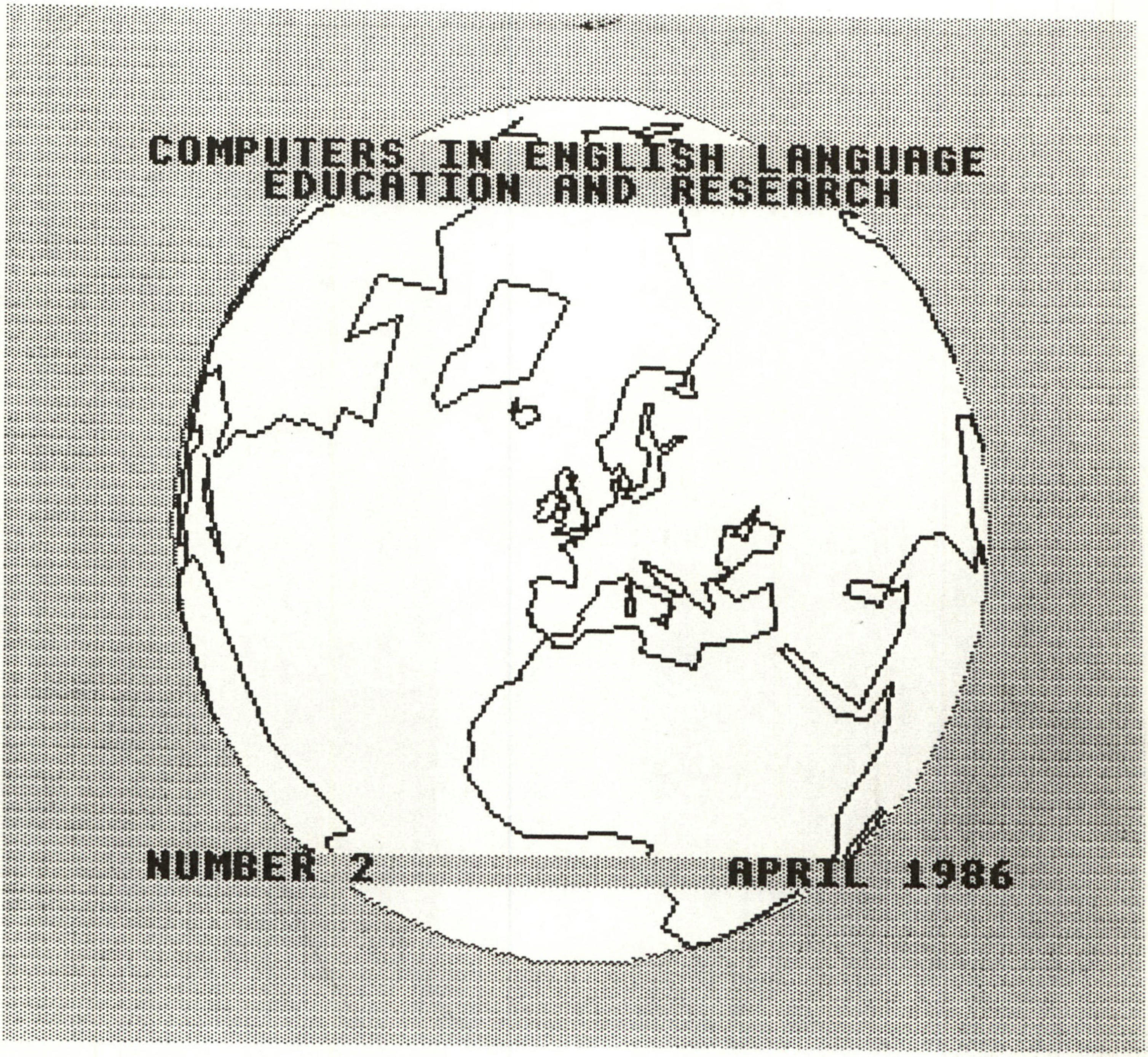
R E C A L L



UNIVERSITY OF LANCASTER

Institute for English Language Education
Department of Linguistics

UNIVERSITE DE PARIS-DAUPHINE

Section d'anglais

RECALL

Computers in English Language Education and Research

Number 2 April 1986

Contents

Published by :

The Department of Linguistics and
The Institute for English Language Education
University of Lancaster
Lancaster England LA1 4YT

Evelyn Perry, Editor
Université de Paris IX-Dauphine
Section d'anglais - B 302
Place du Maréchal de Lattre de Tassigny
75775 PARIS CEDEX 16

Anthony C.M. Cheung
Hong Kong Polytechnic

COMPUTER-BASED ADVENTURE GAMES AND TESL, A SECOND ENQUIRY

I. Introduction

The second phase of a pilot study investigating the effects of adventure games on TESL has recently been completed. Similar to the first phase of the study [1], 'Colossal Adventure', a text-only adventure game was used as the apparatus. However, this time a pair of weak students[2] who were adventure-game novices were used as subjects. They were given a short briefing session in which hints were disclosed on some of the key words that the software understands. In addition, the game objectives were also spelt out. They were then to work on the game and the first session which lasted for an hour was recorded on video.

II. Findings

Although the standard of English of this pair of subjects is lower than that of the pair in Phase I, the findings are nevertheless, quite encouraging in that they are parallel to those of the previous phase :

1. Throughout the session, the subjects were communicating in English. Although in quite a number of instances, their dialogue was very much fragmented, they did not seem to have any serious communication breakdowns. In fact, only one single occurrence of an accidental lapse into the mother tongue was recorded. This is most encouraging, because this pair of subjects represent the kind of students whom English teachers would normally have great difficulties in persuading to interact verbally in the target language.

2. More than 10 instances of giggles and laughter were recorded ; on one occasion, after solving a major problem, one student actually shook the hand of another in delight. The subjects were clearly enjoying themselves.

3. The subjects were found to be consulting the dictionary spontaneously on 6 occasions. This reveals the potential of adventure games as good vocabulary exercises.

4. There were 22 occurrences of the modal CAN. These included 18 occurrences expressing ability (e.g. We cannot go east) which also include 4 occurrences of the sub-category of senses (e.g. We can see a huge spire to the north). The remaining 4 occurrences of CAN were expressions of possibility (e.g. Can I use this word ? (6) jump). This would appear that CAN is also one of the most frequently occurring modals in such interactions, in addition to MAY and MUST which Higgins and Johns (1984)[3] suggest.

III. Conclusion

As can be seen, the findings of this study confirm those of Phase I : adventure games such as 'Colossal Adventure' are good at triggering off oral target language interaction of students. They therefore should be seriously considered as a useful language learning tool. At this point, it seems that a fitting next step is to investigate the effect of move-based simulations on target language attainment.

---

1 A report on Phase I of the pilot study can be found in RECALL, Number 1.

2 The pair of subjects come from the Diploma of Maritime Science Course of the Hong Kong Polytechnic. They attained a bare pass in English in the Hong Kong Certificate of English Examination. A credit attained in this Examination is deemed the equivalent of a G.C.E. O-Level pass.

3 Higgins, J. & Johns, T., Computers in Language Learning, Collins ELT, London and Glasgow, 1984.

David Foulds
Hong Kong Polytechnic

"ROLLING FOG" APPLIED

Is it possible, somehow or other, to measure the level of complexity of text and thereby estimate how easy or how difficult it is to understand ? This question has given rise to much discussion over the past fifty years. Many suggestions have been made as to the factors that need to be considered, and how these can be combined into formulae that will produce some sort of index on a generally recognised scale.

Whether or not we regard readability or 'fog-count' indices as valid, it is a fact that they are used by education authorities, staff training establishments, and by publishers. And language teachers, faced with the problem of deciding which of two texts is the more difficult to read, will often look to some sort of objective reading-comprehensibility measuring device such as the Gunnar Fog-count index to solve their problems.

Computers can do all sorts of remarkable things with text. They make very light work of such things as counting words and punctuation marks, and estimating the length of sentences, and so on. Also, there are sufficient clues in standard English spelling to make it relatively easy to program a computer to identify and count syllables to a very reasonable degree of accuracy. That being the case, the calculation of one of the easier readability indices, such as the Gunnar fog-index, by computer is perfectly feasible.

The Gunnar Fog index was made to function as a pratical, rule-of-thumb guide that could be used by the average teacher with the minimun of inconvenience and difficulty. It is calculated by taking a 100 word sample and, (1), calculating the number of words per sentence in the sample, (2), finding the number of words in the sample which have more than two syllables, and (3), adding the two together and multiplying by 0.4. This is said to give the educational level (in the form of a US public school grade) necessary to understand the sample. So where a piece of writing produces an index of 11, for example, it suggests that the text could be understood by an eleventh grader or above, but not by an average student at a lower grade.

Of course, it is a very approximate measure, and was never intended to be more than that.

The writer has recently completed a package which runs on the BBC-B micro-computer and which calculates the Gunnar Fog-count Index for short texts written on the VIEW word-processor.

Computers, once programmed to work out a formula, can do the task repeatedly without suffering strain, and they do the work very swiftly. It was decided, therefore, when programming this package, not to take single, or even a gapped sequence of Fog index readings of the text, but to take every possible 100-word sample and calculate the indices beginning at the first word and running through the entire text for as far as one could go, - i.e, index (1) from word 1 to word 100, Index (2) from word 2 to word 101, Index (3) from word 3 to 102, and so on. Hence 'rolling" Fog !

The results may be of interest. A number of texts, each about 500 words long, was sampled this way. The product in each case is a mass of figures, each one indicating the readability level for a particular group of 100 words. The immediately observable characteristic of this data is its variability. The following table gives details :

	Average	Highest	Lowest
A.A. Milne (Pooh)	9	12	8
Conan Doyle (Sherlock Holmes)	8	10	6
Swift (Gulliver's Travels)	16	20	14
Thackeray (Vanity Fair)	18	20	17
ELT text (Hong Kong, Primary 4)	6	9	4
Newspaper editorial (SCMP)	14	16	10

Insufficient work has been done to say, for the moment, whether or not the pattern produced by this running calculation of indices is of any great interest in terms of the analysis of text style or structure. The findings, however, clearly show that within a short length of text the Gunnar fog-index can vary as much as 6 points - the difference, it would seem, between a mid-secondary and mid-primary educational level. This suggests that one needs to be very careful indeed in taking and interpreting Gunnar Fog indices. Certainly, to take a single 100 word sample from each of two texts and grade them on the basis of the resultant indices would be most ill advised.

<u>~~Editor~~'s note</u> : Mr. Foulds has offered to send a listing of his program to any interested individuals or institutions.

John Higgins
University of Lancaster

PRACTICAL PARSING

Readers of RECALL may already know I have been awarded a year at Lancaster on a research grant funded by the Leverhulme Foundation. Some of you may be interested to know what I am doing with it.

One of my 1984 Lancaster presentations was given over to ways of communicating with the computer. We looked at the limitations of menu selection, whether by choosing letters or numbers or by stepping through a list using the arrow keys to higlight the next item and RETURN to select the one you want. We looked too at the problems of command-driven systems with word matching, which are found in many educational simulations and, with different effect, in adventure games : GO NORTH, TAKE SWORD, etc. I commented on the frustations of slow keyboard entry. Finally we looked at the rapid whole-phrase entry method used by Robert Ward in his logic problems (which, sadly, do not seem to have been published). We even developed a BASIC routine to do this on the spot, which is included in my second contribution to the book of the course, now in the bookshops.

I am returning to this area, but concentrating on the input of coherent sentences and how the machine is to understand them. Language is, of course, divergent : one cannot predict every possible form that a 'sensible' input can take in the context of a dialogue, let alone a senseless or zany one. To cope with even a part of the range of sensible utterances the machine needs to have a fairly robust parser, not just a matching routine. (The matching routine of ELIZA seems to work after a fashion, but it does not produce proper understanding.) My target this year is to construct limited domain parsers as 'front ends' for several of my existing programs, and to see if I can develop this into a kind of parsing utility which can be used to generate a whole series of such front-ends, or can be turned into an experimental learning tool in its own right. Ideally I would like the 'parser-generator' to be straightforward enough to be usable by teachers and learners with no programming knowledge.

This may seem to be a tall order, but there are several factors which encourage me to think it can be done. The first concerns the semantics. Within the context of a simulation, a game, or a pratice activity, there are only a very few things the machine can do : perhaps change the value of a variable, print a message, or erase and redraw part of a graphic display. This simplifies the semantics of the transaction. In JOHN AND MARY, for instance, the machine's complete knowledge of the world is stored in three flag variables, and, once the machine has identified some input as a command, all it needs to understand is which of these variables has to be flipped. In general, there are only three things we ever want computers to do. We want them to store data. We want them to retrieve data. We want them to process data (and perhaps display the results graphically). These happen to correspond neatly to three sentence archetypes of STATEMENT, QUESTION and COMMAND.

The kind of natural language understanding system I want to create has, therefore, as its first objective to decide if it has been told a fact, asked a question or given an order. The machine itself needs very little further in the way of pragmatics ; it implicitly knows the context of the utterance from the values in its variables (in JOHN AND MARY, for instance; it knows it cannot close the door if the door is already closed) and it does not put itself into an emotional relationship with the user (though the converse may apply). It must, of course, try to co-operate with the user , try to understand as much of the message as possible, even if parts of it are corrupt. It does not need, however, to make wild guesses ; if the data is too corrupt (because it is beyond its parsing range rather than necessarily because it is ill-formed) then the machine can legitimately admit its failure with an "I don't understand" message. Co-operation is two way. In the learning situation we can demand that the learner also try to co-operate in making himself or herself understood. This is the second factor which is in my favour : given the situation of the foreign learner, it is quite legitimate for the machine to be set to handle only well-formed and relevant language, for the machine to be a little bit stupid, hide-bound and literal-minded.

Parsing in corpus linguistics, for which Lancaster is famous, deals with pre-existing text, and therefore permits a choice of strategies between top-down and bottom-up, depth-first or breadth-first, left-to-right or right-to-left. I could give myself the same flexiblility if I use something like the BASIC input routine, so that the machine is given a complete utterance to work on. (This is how JOHN AND MARY works at the moment.) The alternative is to use immediate letter-by-letter parsing, which cuts short an unparsable input before too much time is wasted. This was the approach used in the British Council's demonstration program called FINDER (the one where the suitcases had to be redrawn and coloured in). I am hoping to blend these, using immediate input but storing and displaying unparsable elements in some marked form (such as inverse colour) so that the learner has a clue to whether the machine is understanding the input and, if not, where the problems lie. This will mean using my own pseudo-input routines, but they are not too difficult to write and give me much more control over the display and the timing.

Letter-by-letter parsing can be carried out in a fairly simple brute-force fashion by using a non-recursive transition network which holds the machine's complete recognition vocabulary in a readily searchable fashion. Surprisingly, I have found that the memory costs of this are not high. You need four bytes for each letter in the whole vocabulary plus a small overhead for the array, which means that a 100 word vocabulary (ample for most of the applications I am considering) can be stored in about 2K of memory. I sort the whole vocabulary by alphabetical order, and then create an array (in machine code) which stores the position we have reached in the word, the ASCII code of the current letter, the number of possible successors to that letter, and the position in the array where the first of these successors is to be found. What the program has to do is to accept a letter at the keyboard, search the possible successors to the last letter and see if the new one is legal, and, if so, make the new letter the current letter. If it fails, it inverts the colour of the current word and continues accepting letters until the next space, punctuation or carriage return. It might later be able to say something like "I DON'T KNOW WHAT A WIDNOW IS", or I might give the program a fuzzy matcher to help it deduce WINDOW from WIDNOW. It depends on the nature of the activity.

If the program receives a succession of letters which make up one of its pre-taught words, including a space or punctuation sign or carriage return as an end marker, then it will reach a point in the array where there are no legal successors. This leaves me two spare bytes with which to describe the word. One of these is just an index number, its 'meaning', which the program will be able to interpret as appropriate. The other can be used for a part of speech label, and with a whole byte I can distinguish (if I want to) 255 different parts of speech. I am still playing around with different labelling patterns, looking for the ones which will give me the greatest flexibility later in using comparison tests. I might, for instance, be able to use odd and even above a given range to mark present and past, and thus store all the strong verbs I want to.

However, the end product of this immediate parsing is nothing more than a string of numbers. The machine still has to decide whether the grammar of the string is legal, and then to determine its meaning. The grammar parse can be carried out in exactly the same brute-force fashion as the vocabulary check, but that would be either very limiting or very expensive in memory ; in either case it would be highly inelegant. What I have to do at this point is to create recursive transition networks, in other words groups of these arrays and the ability of one array to call another (or even call itself). That is as far as I have got.

Which programs will I put this parser into ? The first one will be PHOTOFIT, which you may remember is the little game in which you have to study a face and then give orders to a notional 'police artist' to redraw it for you. In this activity all the inputs can be assumed to have the form of commands. There is no need to cope with questions and statements, which simplifies the grammar considerably ; that is why I am beginning with this one. My next project is to put a parser into a new program of mine, tentatively called TIGLET, which is a logic exercise in which you have to offer food to a fussy tiger and try to work out why he accepts some kinds but not others. Here, too, the grammar is simplified since all the user's inputs are offers. With this experience behind me, I hope I will be able to tackle a proper GRAMMARLAND scenario, and the one I have in mind is the corner shop, in which the computer plays shopkeeper to the learner's customer. In this case the learner can ask questions and demand products, perhaps even attempt to bargain. I would be very surprised, however, if I were to get that one finished by September.

I will be glad to hear comments from RECALL's readers, anything from "You're crazy" to "Why not try it this way ?"

Jonathan Upjohn
Université Scientifique et Médicale de Grenoble

MIXED MARRIAGES - INFORMATION TRANSFER AND STORYBOARD

INTRODUCTION

The problems confronting language teachers in the "Université Scientifique et Médicale de Grenoble" are doubtless similar to those to be found elsewhere and thus, the background to the development of the technique to be described is probably typical. We are confronted with the combination of a mixed ability intake, a restricted number of teaching hours (50 per year), and an urgency to teach scientifically relevant material. The solution adopted is also doubtless, a familiar one. It consists in restricting the syllabus by very close definition of the registers, functions and terminal behaviour required. The advantage of such an approach being, that while new skills can be taught to advanced students, at the same time the restricted nature of the task makes it nevertheless feasible for poorer students to attain proficiency.

However, within this framework we found very little CALL material that corresponded to our needs, and consequently this has meant developing various support material. This paper is a description of one of them.

RATIONALE

The material is original (to the best of my knowledge) only in so far as it is a hybrid of two succesful but previously independent techniques. The cross-breeding however, has produced what seems to be a very powerful teaching tool.

The two techniques in question are :

1. Storyboard technique[1]
2. Information transfer technique.

Storyboard is, perhaps, one of the most interesting of all CALL techniques as it has the advantage of being, not only highly motivating but more importantly, the dynamics of the activity, particularly in the later stages are inextricably linked up with contextual implication. That is to say, that the students, without any prior input of heavy linguistic metalanguage are functioning at the crossroads of semantics and structure (morphological, sentential and textual) which is perhaps one of the essential features of human language. And, until a fuller account of learning theory is produced, one can only assume that ensuring high motivational involvement, together with manipulation of language at this level, is a good way of getting the language learnt.

The use of Information transfer technique as the other element in the hybrid has also several advantages :

1. The central role of graphics in science means that information transfer offers a credible work-related task.

2. A judicious choice of graphic prompts makes it possible to isolate just those areas of language which are relevant to the syllabus.

3. As the graphic input circumscribes the area of debate and focuses the students' options, it is possible to upgrade the technical complexity of the text.

DESCRIPTION

A group of learners is provided with a photocopied graphic prompt (statistical table, descriptive figure, graph, flow chart, etc.). The students' task is to discover a target text which is a description of the data. The screen is blank except for numerical figures and for dots, the latter indicating the location andlength of words. Student guesses are typed into the computer and correct answers displayed in context on the screen. The text is thus gradually reconstructed. A help facility, which consists in the display of intial letters, is also provided.

The example given below was designed as an introductory exercise for 4th year university students working in areas of management and applied data processing.

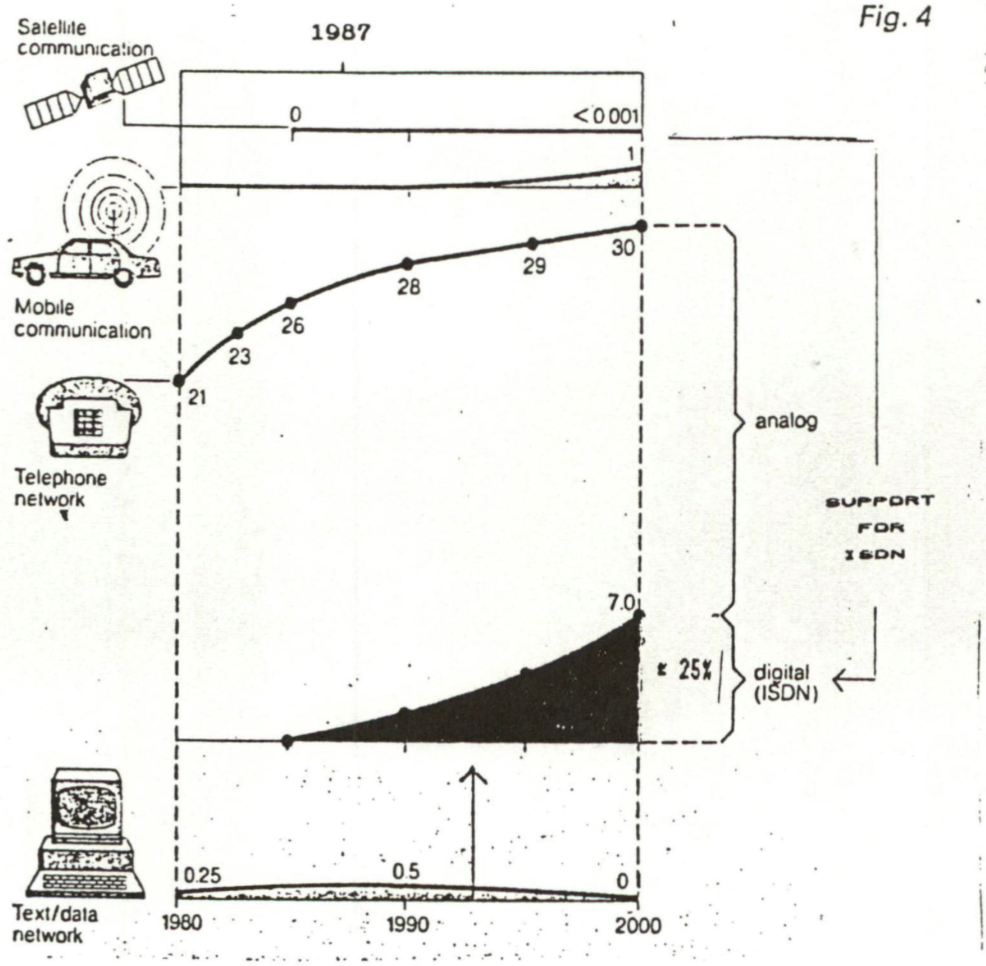
====================

COMMUNICATIONS NETWORK

The trend towards an all-purpose communications network is becoming increasingly clear. That is to say, a network that enables telecommunications for everyone in every desired mode, including speech, text, data and picture simultaneously. Discussion is now focused on the Integrated Services Digital Network (ISDN), which ideally fills the requirements, being adaptable to the existing telephone network which has a high subscriber density and correspondingly low costs. Below, can be seen the estimated increase in the number of subscribers of public networks in the Federal Republic of Germany, expressed in million of line units.

(RECONSTRUCT THE REPORT ON THE DATA CONTAINED IN THE FOLLOWING TABLE.)

Fig. 4



****

The target text proposed for the above figure was the following :

====================

Fig. 4 illustrates the estimated evolution of subscribers to public networks in the Federal Republic of Germany in million of line units.

It seems likely that, over the next 15 years the telephone network will continue to expand from the current 26 million lines reaching 30 million lines by the end of the century. There should be, however, a slight levelling off of the increase rate towards the beginning of the 21st century. This trend will be accompanied by a sharp increase in the rate of growth of ISDN, partly due to support from the communication satelite that is to be launched in 1987, and so, by the year 2000, it is expected that ISDN subscribers will account for roughly 20 % of the whole telephone network. Furthermore, there will be greater demand for mobile communication, with more than 1 million lines by the year 2000.

On the other hand, it is probable that the number of data/text network subscribers, after reaching a peak in 1990 will gradually decrease. Such a decrease will be mainly due to high costs.

====================

DISCUSSION

The notional and structural areas that were designed to be ellicited in this activity can be briefly listed as follows :

- . Cohesive devices. (Furthermore, such...)
- . Compound nouns. (data/text network subscribers...)
- . Passive forms (be accompanied, be launched...)
- . Hypothesis. (it is likely, should...)
- . Collocation. (satelite/launch...)
- . Time relations. (by the end, current...)

It will be noticed that the graph is preceded by a short text. This is not normal storyboard or information transfer procedure. The rationale for such a move is threefold.

1. Whereas a graph only provides conceptual information and thus overt [2] linguistic activity is to be found at the production level only, a prompt text, without decreasing output, also provides an input, thereby enriching the linguistic environment. Hence, it is legitimate to assume that the chances of learning are increased.

2. As has been already mentioned, with reference to graphic input, the increased information at the start of the activity allows one to aim at a very much more complex terminal text, corresponding more closely to the L. 1. technological and scientific skills of the learners.

3. By providing key words, the encoding process can be accelerated. It should be remembered, that it is not profitable for the learner to spend too much time on "blind" guessing as this is a minimal linguistic exercise. Considerable stress has been laid recently on the fundamental importance to language learning of hypothesis testing, but clearly, initial data is a prerequisite for such testing.

The target text can either be extracted from a journal of teacher-written. This makes it suitable for a wide range of teachers. The advantage of teacher-written, or teacher-adapted texts, being that they can be refined into more accurate teaching tools.

The prompt is supplied by photocopied hand-outs. Of course, it would be possible to include graphics in the programme to be displayed on the monitor, but the price of this would be high ; namely it would exclude authoring facilities.

The feed-back that we have had from this technique has, so far, been excellent, (although in fact it has been used less than we would have wished because of access problems) and it is sufficiently flexible to be adapted to many different teaching needs.

---

1 Storyboard is published by Wida Software, London.

2 To say this is not to deny the importance of covert activity, as has been pointed out by R. Allwright (Essex University M.A. seminars, 1975) and others.

BOOK REVIEW

Kurt Moench
University of Lancaster

YOUR CHOICE - Improvised Hackery or Excellence in Software Design

Microcomputers in Education - a critical appraisal of educational software by John Self Harvester, Press Ltd., Brighton, 1985.

Teaching Computers to Teach by Esther R. Steinberg, Lawrence Erlbaum Associates, Hillsdale New Jersey, 1984.

If you are looking for a book that will help you design good software then I recommend the Steinberg book. Teaching Computers to Teach is well organized, easy to read, and it covers the important areas of CAL design : presentation, feedback, user factors, management, diplay, lesson structure, evaluation, and various activity types such as games, simulation, etc. It is a very practical book that is particulary effective in emphasizing the importance of planning and evaluation during each phase of the design process, and in showing you how to do it.

The author's philosophy that the goal of CAI should be to help learners to learn is evident throughout the book. Many examples of both effective and ineffective techniques are provided by the author, who is based at the University of Illinois where PLATO was developed. She also mentions some of the rather limited research findings that are available on CAI, and speculates on their possible implications for software development.

The two areas that the author could have elaborated more on are possible teaching techniques in the CAL lesson, and the classroom implementation. Both topics are covered in what amounts to scarcely a page.

The style and thoroughness of the book also make it an excellent resource for teachers who are getting involved in software evaluation, and want to know what to look for in good courseware. It is also refreshing to see a CAI book that is not simply about programming techniques.

The Self book is "aimed at those who may design and develop educational software in the future", and it does provide incisive analysis of the problems of some of the software currently available both from teachers and commercial software companies. However, he goes much farther and is extremely critical of what he rather deresively call "teachers-cum-programmers", whom he feels lack the knowledge to write good programs (he accuses them of "improvised hackery"), and moreover, have "nothing to write programs about."

According to Self, effective CAI "depends upon the design of good educational software". His approach appears to be "medium centered", rather than learner centered or even teacher centered. This philosophy contrasts markedly with Steinberg who states that the computer (and software) is "merely a vehicle of instruction." Because of this philosophy he seems to be more interested in the educational product than the educational process, and is particularly critical of programs that do not have explicit behavioral learning objectives.

Self concludes that designing good educational software is as difficult as it is important for the future of computers in schools. It should not be left to improvised hackery. He also states that more research, more computer expertise, and more committment to teacher training are required if computers are to be used effectively. All of these are laudable and justified conclusions which make this book worth reading despite its extremely negative tone.

NOTES AND NEWS

Computer News from Lancaster (Geoffrey Leech and Kurt Moench)

. Summer Courses

Last August Scott Windeatt with John Higgins ran a three-week summer course on Computers in Language Learning. The Institute for English Language Education, which organized this course, will be organizing the same course again next summer (17th August - 5th September), and Scott and John will be running the show once more. Looking further ahead, the British Council has asked us to mount a course in the summer of 1987 using the same title as the Course in 1984 (Computers in English Language Education and Research). It looks as if the annual tradition of a computer-language-education summer course at Lancaster is getting well established.

. Computer-based Language Testing

In 1985 the British Council sponsored a project on CBELT (Computer-based English Language Testing) in the Institute for English Language Education. Charles Alderson organized the project, and wrote up the final report. The idea was to investigate new possibilities for using computers in the actual process of language testing (not just in statistical processing of the results). New test types were explored, and both the advantages and the disadvantages of the computer as a testing device were rigorously confronted. One conclusion which seemed relatively uncontroversial was that CBELT and CALL are two sides of the same coin : the conventionally distinct activities of teaching and testing are, from the computer-based viewpoint, refreshingly alike.

. People

John Higgins is spending the year at Lancaster as a Research Fellow sponsored by the Leverhulme Trust. He is working on the concept of "conjectural learning" as applied to CALL, and plans to produce and trial some software before the end of September 1986.

Kurt Moench who is doing the Lancaster M.A. in Language Studies this year is doing his best to turn the M.A. Course (which prides itself on its flexibility) into an M.A. on CALL. He has already started up a Call Users' Group which metts weekly.

Evelyn Perry and her son Martin, now 14 months old, will visit Lancaster for three weeks around Easter 1986.

Gerry Knowles is working on a project for IBM which involves predicting intonation patterns from written texts.

. Available in your neighborhood bookshop now !

Geoffrey Leech and Christopher N. Candlin (ed.), Computers in English Language Teaching and Research,London and New York : Longman, 1986.

Based on lectures and papers from the Lancaster British Council Summer Course "Computers in English Language Education and Research", the book contains papers by Martin Phillips, Graham Davies, John Higgins, John Higgins (again !), Robert Lewis, Annette Odell, Scott Windeatt, Charles Alderson, Jenny Thomas, Gerry Knowles, Geoffrey Sampson, Eric Atwell, John Sinclair, and Geoffrey Leech.

. Unit For Computer Research on the English Language

This research unit (UCREL for short) is shared by the Departments of Linguistics and Modern English Language and of Computing (directors : Roger Garside and Geoffrey Leech).

The Unit has been expanding its research activities, and now has three externally-funded research projects, in addition to John Higgins's CALL project : (1) Automatic grammatical analysis of the LOB Corpus, (2) Development of a context-sensitive spelling error detector and corrector, (3) Development of a corpus of spoken English and the development of corpus-derived pronunciation rules, for high-quality speech synthesis.

We recently had a visit from the world's most advanced speech-recognition group (from IBM), who are interested in a joint research project involving Birmingham, Lancaster, and their own team in New York. The leader of the team, Dr. Fred Jelinek, gave an impressive video demonstration of the current IBM prototype speech recognizer, which can trancribe spoken words from a large vocabulary, but only if the speaker recites the words one at a time, with a pause between each.

OTHER NEWS

. On March 13th - 14th, the Université de Paris-Dauphine, together with the British Council, held a 2-day CALL Seminar at Paris-Dauphine. The seminar was conducted by John Higgins for English language teachers from French universities.

. The CERLACA Research group at Paris-Dauphine, in cooperation with the English Department at the University of Poitiers, is working on transcribing a corpus of (formal) oral English.

. A pilot CALL programme, using primarily text deletion exercises, has begun at Paris-Dauphine in a select number of first-and second-year required English language classes (approximately 25 students per group). Evelyn Perry is coordinating the programme.

. Contributions for RECALL Number 3 should be submitted to the editor (address, p. 2) no later than August 15, 1986. Articles may also be sent to E. Perry, 12 rue Caroline, 75017 Paris, France.