Computer-Assisted English Language Learning Journal

CÆLL JOURNAL

Volume 4, Number 3   Fall 1993

In this issue:

Advertising

Camera-ready advertisements should be sent to the Advertising Coordinator of ISTE. Contact the Advertising Coordinator at the ISTE address below for current information on rates.

Subscriptions

Manuscript Submission

Articles are solicited in all areas having to do with computer-assisted English language instruction. Send articles to the Editor on Apple IIe, MS-DOS, or Macintosh disks in ASCII (plain text) format, accompanied by a formatted paper copy of the article. Graphics can be in TIFF, PICT, or GIF format on disk; a printout of the graphics should be included with the paper copy. References should conform to APA style. Include the author's full name, address and phone number with the article. Authors should be aware that editorial license may be taken to improve the readability of an article or to fit it within the space constraints of the journal.

Research articles should be sent to Douglas Coleman, English Department, The University of Toledo, Toledo OH 43606-3390; email: fac3079@uoft01.utoledo.edu.

If you are interested in reviewing software, please contact the Editor.

REVIEW OF *ADAM & EVE*

Publisher: Oxford English Software, Walton Street, Oxford OX2 6DP, United Kingdom; distributed in the US by Athelstan, 2476 Bolsover #464, Houston TX 77005.

ESOL level: Intermediate to advanced.

System requirements: For the teacher: MS-DOS computer with 512K RAM minimum, hard drive with at least 2.5MB free, monochrome or color monitor. Printer is desirable but not necessary. For the student: MS-DOS computer with 256K RAM minimum, floppy disk drive, monochrome or color monitor. Students do not need a hard drive.

Price: $550 for an individual copy.

The program was reviewed by Vance Stevens, Sultan Qaboos University, on an IBM 8086 with an EGA card and 640K RAM.

One of the most powerful uses of computers in language learning is providing access to massive amounts of text in a target language for pedagogical purposes. Universally machine-readable text is called ascii, which is just plain letters, numbers, and spaces without hidden code. A very large amount and variety of ascii text is available—making programs that can import and use ascii text as grist to a mill among the most useful for language learning. Many currently available programs import such text and then allow students to manipulate or reconstruct the text as part of their language learning experience. *Adam & Eve* takes this activity a step further: it creates a variety of exercises based on prior analysis of the fodder text. In this respect, *Adam & Eve* is a progressive software tool for language learning.

According to Goethals (1987), the project leading to development of this software began in 1979 using a mainframe to concordance and analyse the lexical difficulty of texts and to generate exercises treating target vocabulary, cohesive words, polysemy, and collocations. The project's name is an acronym for Automated Document Analysis and Manipulation capable of producing an Extensible Variety of Exercises. Fifteen years later, *Adam & Eve* has emerged as an example of one thing applied linguistics should do more of for CALL courseware: put text analysis to direct pedagogical use.

Adam & Eve works in two environments: teachers prepare texts and generate materials in master mode, and students use the materials in stand-alone mode. Before teachers can use the program in master mode, it must be installed on a hard disk, and the provided "hardware protection key" must be attached to the computer's printer port. The master program will support separate work areas for as many as 25 teachers. Any student materials created will run in stand-alone mode and may be copied to floppy or hard disk without restriction. The documentation and installation procedures are clearly written and easy to follow, and re-installation easily allows teachers using the software to be added or changed at will. During installation, any teacher may choose to interface with the program in English, French, Dutch, German, or Spanish.

Adam & Eve is intuitively easy to use in student mode, and is not difficult to use in teacher mode as long as the developer has the manual handy. The manual is both comprehensible and comprehensive (basic instructions are given in the first chapters with reference to more detailed instructions to follow). The program itself is menu driven,

with choices made by pressing function keys or moving the cursor from the arrow keys. However, options are not always intuitively apparent, and one would have to be well familiar with the program to be able to interpret all the one-word menu commands. If no on-line help facility is envisaged for the program, it would at least help in future updates to have a one- or two-line explanation of commands scroll in part of the unused screen-space as the cursor was passed over the commands.

In some cases the manual is essential. For example, there is no on-screen menu selection for importing text. One chooses the option "New Text" and then issues the command "i" (for "import") before the name of the text. As the "i" command option is not indicated on the screen, the manual must be consulted beforehand. Furthermore, there is no way of shelling to DOS or getting a directory of files available for import at this point—one must exit the program to go through DOS if a filename is needed.

Another confusing juncture occurs once text analysis has been done. The results must be saved, and the screen suggests "End/Esc = Exit". Although the program convention of using the End key for saving is noted in the documentation, intuition suggests when in doubt press Esc, which in this program means return to the Main Menu without saving, in which case the analysis must be repeated in order to continue. One tends to remember the convention after trial and error. Fortunately, the manual is reasonably well-organized, and it is apparent where to find information and solutions there.

Adam & Eve works on ascii texts of up to 500 lines long, with a maximum of 999 such texts on one system. Before they can be used, texts must be "prepared." For example, none of those 500 lines can be longer than 70 characters long. This is easily handled—the text is simply loaded into a word processor and formatted with margins set to 70 characters (care being taken to ensure that the output is re-converted to ascii). This is an essential step, as characters beyond the 70th on a line are ignored during the import process.

I would imagine most users of *Adam & Eve* would import their text rather than attempt to create it using the in-built text editor. The latter is however possible, and text, once imported (or created) can be edited using the editor provided. This is unfortunately necessary, because one thing the preparer must do is indicate each sentence ending in the text with a carriage return and outdent all lines besides the first in a sentence—in other words, any unique sentence is known to the program only as text beginning in column one and ending with a carriage return. Text can be formatted using the Text Editor, where there is a command TS (Text Split) for finding potential sentence endings. However, format is not automatic—each CR and each outdent must be inserted by hand.

The programmers of this software no doubt felt that it is impossible for a computer program to detect the end of a sentence in 'just-any text' with absolute reliability. The following string illustrates this difficulty:

```
"What is to become of Dr. Hyde?"
screamed Heidi!
```

Although *Adam & Eve*'s assumption (that no full stop, question mark, etc. ends a sentence unless followed by a carriage return) results in accurate analyses, it also means that no text can be used by the program as-is. It also makes it difficult to use texts reformatted by *Adam & Eve* in other applications utilizing ascii text, and may require that users maintain separate versions of their text bases for various applications (having altered one, being sure to alter the other).

Most seriously, the effort involved in hand-entering each CR and space in each outdented line may discourage widespread use of the program in language learning settings, as noted in Stevens (1988). I would rather have seen an approach whereby all strings such as

. ? ! ." .) and so on ...

were considered sentence boundaries. In this scheme, the lesson preparer would have to physically indicate in the text only where these strings were **not** sentence boundaries. In this way, many texts could be used as-is. Furthermore, this approach is consistent with the approach taken toward words in a text that users may wish to have the program ignore in its analysis—for example, proper names that might not be relevant to a linguistic analysis. A greater-than sign inserted in the text before such words renders them invisible to analysis. If a less-than sign were used to annul what the program would otherwise consider to be an automatic sentence boundary, then users might alter the above string as follows:

```
"What is to become of Dr.< >Hyde?"<
screamed >Heidi!
```

Of course, it would be possible to write a preprocessor program in BASIC, C, or another programming language that would prepare the text automatically in the manner required, replacing any of the possible sentence-ending strings with that string plus a carriage return while formatting outdents and 70-character lines as well. Such a program would fly through the text, setting it up the way *Adam & Eve* wants it without further intervention except in the anomalous cases noted above.

For that matter, the publishers could prepare such a program and include it in future versions of their software. As TS finds potential sentence breaks and as there is already a

Verify program to double-check text preparation, it should be a fairly trivial step in the next version to include an automatic preprocessor.

The final step in the text-preparation process is coding the text for rapid generation of cloze, outsider, and sequencing exercises. That the need to code at all might diminish ease of use of this software is acknowledged on p.38 of the manual: "The preparation of exercises which are relevant to needs ... requires some thought and preparation." The coding itself is simple, but there are eight 2-letter commands denoting the beginning and end of text selected for half, full, and blank cloze, sentence and paragraph sequence, and match, scramble, and blank outsider exercises.

Figure 1 shows a portion of a text with added codes and formatting. The Verify function will check for errors in coding, such as a beginning-of-paragraph code with no corresponding end-of-paragraph code or for exercises which are too long for a full cloze.

While it must have appeared trivial to the program authors, this is just the kind of step that average teachers never seem to get

```
.CF SP CH CB OM OS OB
.SS
Adam & Eve works in two environments: teachers prepare texts and
generate materials in master mode, and students use the materials
in stand-alone mode.
.SS
Before teachers can use the program in master mode, it must be
installed on a hard disk, and the provided "hardware protection
key" must be attached to the computer's printer port.
.SS
The master program will support separate work areas for as many as
25 teachers.
.SS
Any student materials created will run in stand-alone mode and may
be copied to floppy or hard disk without restriction.
.SS
The documentation and installation procedures are clearly written
and easy to follow, and re-installation easily allows teachers
using the software to be added or changed at will.
.CF CB CH OM OS OB SP
```

Figure 1. Text with codes for generating exercises

around to (except once, when they want to experiment with how the software works). It would have been better if the software had been configured to make these exercises from the beginning to the *n*th line of any text, with the teacher able to override the default setting by placing codes in the text.

Once the text has been prepared, it can be analysed for sentence and word counts and frequencies. *Adam & Eve* also has a built-in concordancer, creating opportunities for text analysis too numerous and open-ended to discuss in this brief review. The ability to create exercises based on a linguistic analysis is the prime contribution of this program to the field of CALL.

One unique feature of *Adam & Eve* is its ability to assign each word in the text to one of 10 different "sluices." A sluice is simply a way of organizing words. In the sluices provided, the 500 most common words in the language are considered to be in sluice 1, the next 500 most common words in sluice 2, and so on for four sluices up to the first 2000 most frequent words. The user can create word lists for the remaining six sluices, such as different types of English for Specific Purposes vocabulary. In its analysis, *Adam & Eve* registers in what sluice each word in the text belongs.

Once it has been shown what sluices predominate for a particular text, optimal use can be made of that text in the syllabus. For example, students struggling with readings at the 1500 word level would benefit from *Adam & Eve*'s identifying texts that have some sluice 3 words, but few if any at sluice 4. Texts with numerous sluice 4 words would likely be too difficult for such students, and texts with few sluice 3 words too easy. But a text with a certain number of sluice 3 words would be just right because

Adam & Eve can target those sluice 3 words, creating exercises focusing on exactly the vocabulary the students need to learn. The same can be done for specialized vocabulary, using the sluices whose contents have been defined by the user.

The ability to use analysis to generate a wide variety of exercises, either hard-copy or on-line interactive, distinguishes *Adam & Eve* from other software packages for language learning. Besides the different cloze and sequence exercises mentioned above, *Adam & Eve* can create outsider exercises which use words outside a given sluice as targets for deletion in cloze exercises, narrowing the focus of language practice on just those words.

Polysemy exercises work from a database in the software—when a word with multiple meanings is detected in the text being used, the examples and exercises presenting all the possible meanings will be created from the database. Other exercise types are Verbs, in which the infinitive is given for each verb in the text so that students must restore the correct tense, and Cohesion, which treats prepositions, conjunctions, and pronouns. Collocation exercises are also possible on creation of an appropriate database (database creation entails scanning text manually, highlighting collocates, and translating or paraphrasing them; perhaps OUP will eventually prepare a collocation database to share with registered users of *Adam & Eve*).

The on-line exercises, while professionally presented, could be improved in manner of feedback. Many of the exercises have students type numbers as answers, and the numbers are inserted in the blanks with no further feedback until F1 Solution is pressed. Even then, students never see words

> The ability to create exercises based on a linguistic analysis is the prime contribution of this program to the field of CALL.

in the blanks, only numbers. Also, the program doesn't always tell students once they have answered a question if they are right or wrong; rather, it may recycle questions missed. Such considerations are minor relative to the potential of generating directed practice from target texts, but feedback is something Oxford might look into in preparing version 2 of the product.

A pedagogical decision of another sort resulted in the following problem: When I first created a student disk and gave it to my students, it worked well initially, but the next student who tried it couldn't run it—the program emitted frustratingly enigmatic bleeps when attempts were made to retry the lessons. It turns out that by default, a student disk must be used in one sitting (to prevent students seeking "solutions from other parts of the program," Manual, p.97). The rationale for this is that because *Adam & Eve* can be used for delivering "formal tests," second attempts to use student disks are blocked.

In order to let as many of my students as possible explore the program in self-access, I had to quickly find in the manual how to run the executable module with a parameter which sets the disk back to zero (wiping out previous interaction data, but at least the disk can then be used by any number of students).

In these and other instances noted above, I find that

(1) the default settings of the program do not match everyone's pedagogical instincts,
(2) the program is versatile enough to provide over-rides to the default settings, but
(3) not just any teacher is going to be able to find and implement these over-rides, which could result in frustration, and
(4) the program would therefore probably be best utilized in a setting where one teacher or systems manager became familiar with using the program and helped others to use it properly.

Language teachers cognizant of the potential of text analysis for language learning will be excited about the prospect of using a program like *Adam & Eve*. In actual use, one finds many areas where the program could be improved, and this review has in part been directed at encouraging the publishers to keep working at producing an even better version of their product. While this review has touched on such areas, the overall opinion of this reviewer is that this software is a remarkable product which charts desirable directions in the development of language learning software.

Time spent in coming to grips with *Adam & Eve* should be worthwhile in terms of benefits to students and for insights gained in applications of text analysis to language learning. Oxford has done a fine job on program presentation, and use of the software is highly recommended. No DOS-based CALL lab should be without a copy of *Adam & Eve*.

For more information, contact Vance Stevens, Language Centre, Sultan Qaboos University, Box 32493, Al-Khod, Sultanate of Oman.

REFERENCES

Goethals, Michael. 1987. *Automated document analysis and manipulation and extended vocabulary exercises: From the mainframe to the PC, the way to the teacher's electronic desk.* Paper presented at the AILA 8th World Congress of Applied Linguistics, Sydney, August 16-21, 1987.
Stevens, Vance. 1988. A critical but overlooked factor: Not can but will teachers modify the software? *C.A.L.L. Digest,* 4 (8), 3-5.