A Computerized Comprehensible Writing Aid

FINAL REPORT

David E. Kieras

University of Michigan

Report No. 27 (FR-87/ONR-27)

September 28, 1987

This research was supported by the Office of Naval Research, Personnel and Training Research Programs, under Contract Number N00014-85-K-0385, Contract Authority Identification Number NR 667-547. Reproduction in whole or part is permitted for any purpose of the United States Government.

Approved for Public Release; Distribution Unlimited
This is the final report for a research contract concerned with further development of a computerized aid for preparing comprehensible technical documents. This contract was a follow-on to the contract N00014-84-K-0729, NR 667-513. The project will be further continued as a supplement to ONR contract N00014-85-K-0138, NR 667-543.

The goal of this project is to develop a computer program that will serve as an editorial tool for improving the comprehensibility of technical documents such as training materials or technical manuals for equipment. The program would take as input a draft document and would output a sentence-by-sentence critique of the comprehensibility of the document, identifying problems and suggesting solutions where it is feasible to do so. This final report provides a summary of the current state of the project.
This is the final report for a research contract concerned with further development of a computerized aid for preparing comprehensible technical documents. This contract was a follow-on to the contract N00014-84-K-0729, NR 667-513. The project will be further continued as a supplement to ONR contract N00014-85-K-0138, NR 667-543. The goal of this project is to develop a computer program that will serve as an editorial tool for improving the comprehensibility of technical documents such as training materials or technical manuals for equipment. The program would take as input a draft document and would output a sentence-by-sentence critique of the comprehensibility of the document, identifying problems and suggesting solutions where it is feasible to do so. The work under current funding is intended to produce a prototype system that runs well enough for writers to try out in an actual working environment, even if the system is relatively limited.

The work on this project has been slow and difficult. The final report (Kieras, 1985c) of the previous contract summarizes the previous effort on this project. This final report provides a summary of the current state of the project.

WORK ACCOMPLISHED

Comprehensibility System Development

In this project, three separate comprehensibility systems have been developed and implemented. For purposes of providing background, the summary starts with the first, developed under the previous contract.

Demonstration system. This first system is described in Technical Report No. 17 (Kieras 1985a). This system was originally implemented in UCI LISP and ported to the INTERLISP-D environment running on an 1108. The function of this system was to demonstrate the concept of an advanced aid for comprehensible writing. Although assembled out of existing components, this simple system could generate much more sophisticated criticisms of comprehension problems than existing writing aid software.

This system was based on the simulation model of comprehension described in Kieras (1983). The basic organization
of the demonstration system has been retained in its successors. Figure 1 shows this structure as currently implemented. The augmented transition network (ATN) parser analyzes the surface structure of the input sentences and produces the corresponding semantic structure for the sentence content in the form of an ACT semantic network (Anderson, 1976). This semantic structure is tagged with information about the corresponding syntactic form in the input. For example, the main proposition is tagged if it appeared in the passive voice. An example appears in Figure 2.

A reference resolution module then examines the portions of the network that correspond to noun phrases and then compares them to the semantic content of the preceding text to determine what the referent is for each noun phrase. This defines what information is "given" versus "new" in the sentence; after all other processing of the sentence is completed, the new information is added to the semantic structure for the passage.

After reference resolution is complete, a criticism module, consisting mostly of production rules, analyses the pattern of syntactic tags on the semantic structure with regard to the previous text content, and generates criticisms or comments. For example, if the main proposition of a sentence was stated in the passive voice, but the referent of the surface subject noun phrase is not also marked as the current discourse topic, a criticism is produced that the passive voice has been used improperly. Figure 3 gives examples of such rules, using the current PPS rule representation.

Graphic interface system. The second system was begun at the end of the previous contract and occupied much of the time in the present contract. This system was a response to the fact that the main problem that had to be solved in order to develop a usable comprehensible writing aid was the extension of the parser in the demonstration system. The object of the graphic interface system was to take advantage of the INTERLISP-D environment by providing direct manipulation and display facilities (a "parsing workbench") that would allow us to develop ATN grammars much more easily than previous programming environments had permitted.

The natural form for expressing an ATN grammar is as a network, consisting of nodes that represent parsing states interconnected with arcs that denote what kind of lexical item or grammatical construction must be present before the parsing can go to the next state. The graphic interface system actually displays these networks directly in a large window, and, by selecting nodes or arcs with the mouse, the grammar developer could view, edit, or modify these nodes and arcs. In addition, the grammar could be tested easily. When a parse was underway, the display would contain the parsing network that was currently active, and nodes and arcs in the network would be highlighted to show the current parsing state. The parse could be speeded up, slowed down, or halted in midstream to allow inspection of the
Figure 1. Current structure of the comprehensibility system.
The energy booster is used by the phaser system.

Figure 2. Example of syntactically-tagged semantic structure for a sentence.

(BadPassive
  IF
    ((CONTROL GOAL DO CRITICISM)
     (SM TAG ?MAIN-PROP PASSIVE)
     (SM TAG ?MAIN-PROP STATEMENT-MAIN)
     (SM TAG ?STATEMENT-SUBJECT STATEMENT-SUBJECT)
     (NOT (PM TAG ?STATEMENT-SUBJECT DISCOURSE-TOPIC)))
  THEN
    ((PRINT-MSG "BAD PASSIVE")))

(GoodPassive
  IF
    ((CONTROL GOAL DO CRITICISM)
     (SM TAG ?MAIN-PROP PASSIVE)
     (SM TAG ?MAIN-PROP STATEMENT-MAIN)
     (SM TAG ?STATEMENT-SUBJECT STATEMENT-SUBJECT)
     (PM TAG ?STATEMENT-SUBJECT DISCOURSE-TOPIC))
  THEN
    ((PRINT-MSG "GOOD PASSIVE")))

Figure 3. Example of criticism rules in PPS format.
current state or results of the parse. Figure 4 is an example of the ATN display. Other important facilities were also available, such as a graphic display of the tagged semantic structure produced by the parser, illustrated in Figure 5.

The overall quality of this implementation was very good, but it turned out, for reasons that will be described below, that even this powerful development environment was not adequate to the task of developing a large, complex, ATN in a reasonable amount of time. However, the code is still available and will be supplied on request.

Using the graphic interface system, an effort was made to construct a large ATN adequate to handle samples of actual draft technical writing supplied by NPRDC. An attempt to make use of previously published ATNs or large grammars was not very fruitful, because they had not been developed with this type of prose in mind.

A complete version of a comprehensibility system was assembled in the graphic interface package, and demonstrated to NPRDC personnel on two separate visits to our laboratory. The input sentences would be fed one at a time either from the 1108 keyboard or from a file. The output of the system was comments on each sentence, expressed, as in the demonstration system, in terms of psycholinguistic jargon.

As in the demonstration system, the ATN parser automatically generated an ACT representation of the sentence content as the sentence was parsed. A rather large lexicon of military terms, supplied by NPRDC, was incorporated. As an experiment, the reference resolution module consisted of a set of production rules to resolve noun-phrase referents. This resulted in slow processing, but this approach is potentially more powerful than alternatives, and so might be good for the more subtle and complex forms of reference that might need to be handled. A few criticism rules were implemented using the PPS system (Covrigaru & Kieras, 1987) developed under other projects; the example rules in Figure 3 are from this system.

The graphic interface system was far too slow, even with the display generation turned off. A major speed obstacle with the system was the parsing, followed by the relatively low speed of the production system interpreter. This slow execution could be largely blamed on the Xerox 1108, which is not at all a fast computer, together with the fact that full advantage was not taken of some of the possible ways to speed up ATN interpreter, and also on the fact that the noun-phrase resolution was handled in a rather slow fashion as well.

Grammar development system. The difficulties alluded to above of developing a large ATN grammar turned out to be quite
Figure 4. ATN display in the graphic interface "parsing workbench" system. The stack at the right shows the current network embeddings, while the major noun-phrase ATN appears as the network in the main window.

Figure 5. An example semantic structure display in the graphical interface system.
serious, even with the powerful graphic interface system. The reasons for this will be more clear from a summary of the grammar development process: Based on the NPRDC samples, we would choose a particular syntactic pattern to be included in the developing grammar. Modifying the grammar to allow for the parsing of this pattern would then consist of adding various new arcs and nodes to the ATN. This had to be done in such a way that it would not disturb existing parsing pathways. In addition, code would have to be written to produce the semantic structure denoted by the syntactic pattern, and incorporated into the ATN. These details could be handled with enough care and testing, but as the grammar grew in complexity, the time required became increasingly long.

In response, John Mayer, a graduate student who performed the detailed grammar development, developed a specification language for ATN grammars which then made it possible to relatively quickly develop a large grammar using the NPRDC samples. This work is described in Technical Report No. 25 (Mayer and Kieras, 1987). This specification language has similarities to BNF, standard linguistic rewrite rules, and regular expression notation. Instead of specifying which nodes should be connected with which arcs, the grammar developer can express a new pattern very compactly in a linear notation, and the specification language compiler would then generate the corresponding new ATN. (See Figure 1.)

In the time available, Mayer developed a compiler for the specification language, an interpreter for the ATN produced by the compiler, and a large grammar for technical prose. This ATN interpreter produces a syntax tree, rather than a semantic representation, so its use in the comprehensibility system will require developing a syntax-tree to ACT representation translator. The coverage of the large grammar seems good. In the middle of development, using a series of samples from NPRDC, this parser could handle at least 78% of the sentences in the samples, and at the end of development, as many as 96% of the sentences in a new sample. The judgement was made in Technical Report No. 25 that this parser had adequate coverage to be of practical utility in the comprehensibility system.

The full documentation of this work is contained in Technical Report No. 25, which contains a formal definition of the specification language, the algorithms for the compiler and parser, a description and specification of the large grammar that was developed, and results on the syntactic coverage of the grammar using the NPRDC samples.

Linguistic Analysis Work

A small amount of support from this contract in the form of graduate student support was used for some linguistic analysis that supported the overall goals of this project. This work was
done by Leslie Olsen, with the assistance of Rod Johnson, a linguistics graduate student. The work was an analysis of passages collected by Bruce Britton from Army sources (Britton & Glynn, submitted). These passages appeared in both an original and revised form as examples in an Army document on how to revise technical materials. Britton had conducted some recall studies showing that the revised versions of the passages did result in better recall of the information after a delay. Britton supplied us with both versions of these passages. The question was what the improvement in recall was due to. One explanation suggested by Britton was that the improvement was due to the revisions using various signalling techniques such as improved headings and typefaces. The analysis conducted by Olsen and Johnson compared the versions of a few of the passages in terms of referential forms and cohesion.

The results were that the revisions were as good or better than the originals both in terms of referential use and cohesion; thus the improvements could be due to these factors, rather than the others suggested by Britton. One major difference was that the revised versions replaced telegraphic forms of noun phrases with full ones as well as fixing certain other referential problems (the telegraphic style is common in the NPRDC samples).

Another difference was that many procedures were presented in a descriptive way in the originals, but in an imperative form in the revised version; thus a lack of focus in the descriptive form was replaced with a fairly sharp focus around the chain of imperatives in the revised version. Furthermore, the value of the procedural content being expressed in the imperative corresponds to remarks by Kieras (1985b; in press) about the benefits of being able to construct production rules easily from text.

This work, which will be written up in the future, suggests that some of the criticisms of the system will be valuable.

Summary of Work Accomplished

In the period covered by this report, a substantial attempt was made to develop a prototype of a practical comprehensible writing aid. Developing an adequate parser was more difficult than anticipated, not due to a lack of technical or scientific concepts, but rather to the sheer bookkeeping problems of trying to develop a large set of parsing rules. As a result it was not possible to develop the full prototype system. Rather, a complete system was developed that had too limited a grammar to be usable, followed by development of a tool that enabled development of a full sized and usable parser, but additional work is required to produce a complete comprehensibility system. Finally, a small amount of linguistic analytic work, together with the earlier experimental work done on this project, shows
that the kinds of problems that the system could detect are important qualities of more comprehensible technical documents.

REPORTS PRODUCED


REFERENCES


END
12-87
DTIC