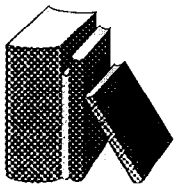

Book Reviews

Machine Translation: Past, Present, Future

Nigel Ward

Machine Translation Past, Present, Future (Ellis Horwood Limited, Chichester, England, 1986, 382 pp., $52.95) by William John Hutchins is a good book. It is organized around projects as "a history and assessment of efforts to mechanise processes of translating" (p.18). It is complete, discussing basically every project in the world since machine translation's first glimmerings 40 years ago Projects are grouped by time frame, nation, or approach. The organization is, of course, somewhat arbitrary, but it is supplemented by cross-references and summary tables of projects and systems. Hutchins not only presents the theories, algorithms, and designs but also the history, goals, assumptions, and constraints of each project. There are many sample outputs and fair evaluations of the contributions and shortcomings of each approach. Hutchins also includes thorough chapters giving background and commentary on problems, strategies, and methods; for example, the discussion of the ALPAC report includes sketches of the committee member's backgrounds. The writing is generally clear and is well supported by figures

This book is a fine reference, with an extensive bibliography and a good index. It is also a readable introduction to the field. It assumes no previous knowledge of machine translation or even programming. However, because the book includes no traces of system executions, it is harder to get a feel for how these programs work. The book has a few other imperfections as well It has numerous typos. The completeness is obsessive; a reference exists to every known machine translation project, no matter how obscure or unimportant. Hutchins sometimes loses his (generally healthy) critical distance, especially for the recent projects In particular, he sometimes gushes enthusiastically in what appear to be paraphrases of press releases and special pleadings.

Readers used to AI writings might find this book unusual. There are no mordant criticisms of misguided research. Instead, Hutchins notes the good points and shortcomings of every project. As an outsider (a librarian), Hutchins apparently has no axes to grind. Also, he makes no overly simplistic generalizations. Hutchins presents the common dichotomies of machine translation research (such as practical versus theoretical, empirical versus perfectionist, and direct versus indirect) but uses them only to clarify. Many surveys have cast machine translation research as a story of progress toward the ultimate design, for example, lexical to syntactic to semantic. Hutchins offers instead an intellectual history of the field, pointing out how ideas mutate, combine, and reemerge in new contexts. For example, a recurring theme is the notion of "interlingua." Hutchins thoroughly presents the variations on this theme, including "the centuries-old notion of a 'universal language'" (p 129), the idea of "a basic universal set of primitive sense elements" (p. 129), the idea of a "common 'deep syntactic' base" (p. 190), the idea of "a direct representation of 'conceptual relations' independent of any language" (p. 129), the idea of an interlingua based on "the sum of all features common to two or more languages" (p. 129), and the idea of "a complete artificial language with its own lexicon and syntax" (p. 129) Hutchins also explains how these various notions of interlingua are reflected in implemented systems; cites Booth's observation that all source-language to target-language conversion involves abstract representations which could be regarded as intermediary; and notes that "the differences between the 'interlingual' and the 'transfer' approaches have become blurred in Logos, Eurotra, and METAL" (p 323).

How does AI relate to machine translation? Hutchins is not convinced that AI experiments can be expanded into large-scale systems. It is not clear whether AI ideas such as scripts and decompositions into primitives provided much leverage on the problems of machine translation It is also not clear that the AI philosophy of understanding and meaning (p 327) provides a useful or original theory of machine translation. Instead, Hutchins foresees "a gradual integration of AI approaches" into machine translation systems (p. 329). It is sobering to note that the most widely used machine translation system, Systran, is remarkably kludgelike and theory-free. It is the result of hundreds of man-years of work—mostly on dictionaries—over three decades. Hutchins concludes, "The lesson from the past ought to be clear enough: the complexity of MT derives from the complexity of language." (p. 333).

Nigel Ward is a post-graduate researcher in the Computer Science Division, University of California at Berkeley, Berkeley, CA 94720

A Comprehensive Guide to AI and Expert Systems: Turbo Pascal Edition

Gregory B White

A Comprehensive Guide to AI and Expert Systems Turbo Pascal Edition (McGraw-Hill, Inc., New York, 1988, 257 pp , $19.95) by Robert I. Levine, Diane E Drang and Barry Edelson might better be titled "An Introduction to Expert Systems: Turbo Pascal Edition." Emphasizing expert systems, the book did not live up to the promise implied in the title. In addition, claims on the back cover stated that the book "illustrates how anyone can create his or her own expert systems on a home/personal computer." Yet, when I completed the book, I found myself doubting that it was possible to create an expert system in Pascal Finally, the preface stated that the "book is designed to be used by advanced high school students, college students, corporate presidents who want to understand artificial intelligence, and curious noncomputer people." This somewhat less lofty goal was also not reached because it would take an individual who had already mastered the basics of Pascal to understand the programs contained in the book In short, I found this book to be too incomplete to be considered an introduction to AI, too complicated to be used by nonprogrammers, and too simplistic in its approach toward the few basic AI concepts presented to be used as a textbook for any AI course.

The first four chapters of the book are an attempt to introduce the reader to the field of AI In these chapters, the authors attempted to lay the foundation for later discussions on specific areas of AI They introduce such topics as chaining, blackboards, pruning, and heuristic rules but rarely spend more than a few paragraphs on any of them. They leave the reader with the impression that AI is the panacea for all programming problems Statements such as the following do nothing for the field of AI except cause more false expectations and probably leave the reader worse off with this little bit of knowledge than with none at all

> Artificial intelligence, as its name implies, really does enable a computer to think By simplifying the way programs are put together, AI imitates the basic human learning process by which new information is absorbed and made available for further reference (p. 2)

> A standard program can do everything an artificial intelligence program can do, but it cannot be programmed as easily or as quickly (p 4).

The authors repeatedly present some material and give an example of what they are trying to convey, but they stop before successfully imparting some knowledge on the subject and its relationship to AI For example, in chapter 2 they discuss a program to determine if a child has learning problems in arithmetic. The authors begin by explaining some of the rules that would be used to indicate a learning problem existed (such as "can the person subtract two numbers with single column borrowing"). This one example can be used—and has been—to show how an AI approach might differ from a traditional approach Generally, the discussion centers on how a traditional approach might be to simply give a student a number of problems to solve and then report on the number of correct versus incorrect responses The AI approach (as seen in the field of intelligent computer-aided instruction) would analyze the problems the student missed in order to see if a pattern existed (such as not being able to subtract two numbers with single-column borrowing) The program could then concentrate on this area and provide some additional instruction to help the student overcome the difficulty This example would have lent itself well to the structure of this book because it would have been easy to create a simple program to present subtraction problems to a student This small program could then be added on to build a more intelligent system by incorporating the borrowing rule and other elements, such as weighting factors. Our authors, however, simply wrote a program that literally asked whether a student had problems, as follows:

```
WRITE('CAN ', PERSON,' SUBTRACT TWO NUMBERS WITHOUT BORROWING?');
READLN(S1);
WRITE('CAN ', PERSON,' SUBTRACT TWO NUMBERS WITH SINGLE COLUMN BORROWING?');
READLN('S2);
```

This sample program does not impart any information or wisdom about expert systems or AI It is not what I would call an expert system. The authors then go on to make a clumsy attempt at introducing weighting factors and never again discuss the subtraction problem By not making better use of this example, they missed a tremendous opportunity to easily explain some AI concepts to novice learners

The second section of the book, chapters 5 through 7, is an introduction to some fundamental concepts used in expert systems and knowledge engineering. Chapter 5 serves as an introduction to forward chaining; backward chaining is discussed in chapter 6. In both chapters, the authors did a respectable job introducing the topics but failed to pick a suitable example and create a corresponding Pascal program For example, in chapter 5 they create a program they claim can be changed to accommodate another topic by replacing the knowledge base. This task is not simple and basically means rewriting the entire program. This programming example would be especially upsetting to a novice If an individual followed the instructions to create and insert a new knowledge base, the program would not work The supposedly step-by-step process simply did not work because the authors did not explain all the changes that would have to be made.

The backward-chaining example in chapter 6 seems more appropriate for forward chaining They attempt to build an expert system to determine if an applicant is qualified for a job However, they begin by alluding to a program to help discover why a car won't start. I do not understand why they did not continue to develop this backward-chaining program; the general public can easily relate to the car problem For their job applicant prob-

lem, they built a decision tree.

Chapter 7 introduces probability theory and fuzzy logic, and the authors did a good job of introducing both topics at a level the audience could understand.

Section 3, chapters 8 through 10, consists of three expert system examples. I believe the goal is to attempt to apply the principles learned in the previous sections It was here I began to lose all confidence that it is possible to write an expert system using Turbo Pascal. The examples were all so difficult that I couldn't see how a novice to AI could actually grasp the principle needed to understand what an expert system is. By making some giant leaps in logic, one might understand the instruction, but they could have chosen some better examples and slowly developed them so the learning could have been almost effortless. It seemed the authors' implied definition of an AI system is that any amount of knowledge, no matter how small, makes a program an expert system. It would have been better for them to remember the definition-description of AI that Elaine Rich proposed in her book *Artificial Intelligence* (New York: McGraw Hill, 1983): "Artificial Intelligence (A.I.) is the study of how to make computers do things at which, at the moment, people do better" (p. 1).

The remaining three sections and eight chapters introduced topics such as object-oriented programming, semantic nets, certainty factors, automated learning, Lisp, and Prolog. Once again, I found the topics were covered in a shallow manner, and the programming examples were ineffective. The two chapters covering Lisp and Prolog, however, were covered in sufficient detail for the intended audience. I would have liked to have seen a little more information about why these languages are the "languages of choice" for AI programmers. In other words, it would have been nice if the authors had presented enough material so that a reader could answer the question, "Why do people program in Lisp or Prolog?"

To summarize, I did not find the book adequate as an introductory textbook or as information for the curious about what AI is or what makes a program an expert system. If the authors ever produce a second edition (and attempt to make their text a true "Comprehensive Guide to AI and Expert Systems") I would recommend the following changes· First, include information on other fields in AI such as pattern recognition, intelligent computer-aided instruction, gaming, problem solving, and robotics, and expand the discussions on topics such as goals, pruning, blackboards, and inference mechanisms. Second, include a section on current and historic expert systems to give users a better feel for exactly what they can and are doing. Third, rewrite the programs and do a better job of building on previously discussed ideas. Find examples the intended audience can better understand and relate to (such as the traditional programs for selecting wine, troubleshooting automobiles, or diagnosing an illness). Show how these examples can be real expert systems and not just academic exercises.

U S Air Force Captain Gregory B White works as chief of the Networks and Computer Systems Branch at the U S Air Force Computer Security Office

Expert Systems for the Technical Professional

Francis D. Tuggle

Expert Systems for the Technical Professional (John Wiley and Sons, New York, 1987, 314 pp., $32.95) by Deborah D Wolfgram, Teresa J Dear, and Craig S. Galbraith is a reasonable book to read for an appreciation of how to develop an expert system. The authors are all connected with the Graduate School of Management at the University of California at Irvine; Galbraith is on the faculty, and Wolfgram and Dear received their Masters of Business Administration at the school. This fact is germane because the book is strongly oriented toward the manager, as opposed to the technical professional, in spite of the book's title. However, one can make allowances for this fault. The book touches on the important topics, it is complete, and it is meaningfully organized for the novice. Thus, an intelligent manager who wants a step-by-step, practical guide to the construction of an expert system can benefit from this book A technical professional who is completely ignorant of expert systems can find this book a useful introduction, but this person's appetite for detail will not be completely whetted.

The book occupies a pleasing middle-ground position when compared to others on expert systems; indeed, for the managerial professional, it is one of the best. In a sense, this book can be meaningfully compared to an orderly expansion of section three, Building an Expert System, of Waterman's text (*A Guide to Expert Systems*, Reading, Mass Addison-Wesley, 1986). I grant you this book is more technical than Chorafas's general guide to the business expert system (*Applying Expert systems in Business*, New York: McGraw-Hill, 1987) and is far more technical than Hertz's treatise on expert systems and executive information systems (*The Expert Executive· Using AI and Expert Systems*, New York: Wiley, 1988). However, it is considerably less technical than *Building Expert Systems* by Hayes-Roth, Waterman, and Lenat (Reading, Mass.: Addison-Wesley, 1983) and *Business Expert Systems* by Holsapple and Whinston (Homewood, Ill.: Richard D. Irwin, 1987)

Holsapple and Whinston stress an in-depth development of one expert system using a single shell in order to reveal technical details, and Hayes-Roth, Waterman, and Lenat relate snippets of various expert systems, languages, and shells in order to lay out considerable technical detail about the construction of expert systems. Wolfgram, Dear, and Galbraith, however, avoid the monomania of the Holsapple and Whinston book by presenting multiple languages, shells, and hardware configurations and the jumpiness of the Hayes-Roth book by providing a complete, well-structured methodology for developing an expert system from scratch.

The virtues of this book are concentrated in chapters 7, 8, and 9 on knowledge engineering, knowledge

acquisition, and five different commercial expert system development tools (1st Class, Personal Consultant, EXSYS, GURU, and KES) applied to the same problem If the reader is inexperienced with the subject, useful information is also available on various knowledge representation schemes (semantic nets, frames, production systems, first-order predicate calculus, and so on); two problem representations (state space and problem reduction); a variety of problem-solving methods (search—blind and heuristic—versus control versus reasoning); hardware (conventional machine architecture versus symbolic processor versus connectionism); and software (AI languages versus shells versus prepackaged commercial expert systems). These topics are sketched in chapters 3 through 6.

Given the large number of topics the authors cover, there is neither much depth nor rigor to the concepts. For example, I find it hard to believe that anyone can have much use for a 2-1/2-page introduction to first-order predicate calculus, not to mention other important ideas presented for the first time in a similarly short manner Perhaps such a short review will stimulate an unknowledgeable reader to look for detailed coverage elsewhere—I sure hope so.

The authors' penchant for brevity carries them to short descriptions of expert systems used by various organizations (Westinghouse Electric Company, Federal Aviation Administration, Campbell Soup Company, the insurance industry, Infomart, and so on) and has them providing eight pages of single-line listings of applications (media buying, deep space station designer, database management system selection, Pascal program debugging, carburetor fault diagnosis, and so on). Who derives any benefit from such sketchy information? Is the reader supposed to get a spark of insight from the extensive listing of project titles about where expert systems might be productively developed for the organization? If so, the reader is going to need more information about the projects or citations to publications about them—neither of which is provided

The chapter on building an expert system using five different commercial systems also suffers from this tendency to be short-winded. The driving example is that of a personnel officer seeking to hire someone for a company position (which also does not do much to reinforce the supposedly technical orientation of the book). The personnel officer uses two rules of thumb to guide hiring an applicant (1) if the person's attitude is good and the references are good, hire and (2) if the attitude is good but the references are poor, the applicant must have at least an adequate education and one year of experience in a similar position As a consequence, the reader has no feel for the problems in constructing and managing a large rule set and the difficulty in coping with experts with partially contradictory rules, finding a good firing order for the rule set, and so on However, the reader does get a good sense of the expressive power, usability, and feel of these five commercial expert system shells, which might be enough for the reader to correctly infer which is most suitable for the application and the organization.

The book also contains the usual set of annoying typographical errors: Cadauceus for Caduceus, Project MAX for Project MAC, University of South California for University of Southern California, Travellors Corporation for Travelers Insurance, and so on.

Francis D Tuggle is the Jesse H Jones Professor of Management at the Graduate School of Administration, Rice University, P O Box 1892, Houston, TX 77251

Natural Language Understanding

Dekai Wu

The jacket notes to *Natural Language Understanding* (The Benjamin/Cummings Publishing Company, Inc., Menlo Park, Calif., 1987, 574 pp. $31.96) by James Allen, say it is "the most comprehensive, in-depth book to date covering all major aspects of natural language processing " This claim is probably realistic, especially given the limited selection currently available. The unsettled state of natural language as a research area makes it difficult to evaluate a text that purports to be comprehensive. It is tempting to make facile statements about lack of depth, spotty coverage, and poor organization; however, these criteria are for a mature field. As it is, everyone is sure to disagree about the specific shortcomings, and simply to make such criticisms would be to miss the main point: Allen has indeed done an excellent job of putting together a wide survey of techniques used in computational linguistics

The book is divided into four parts: Syntactic Processing, Semantic Interpretation, Context and World Knowledge, and Response Generation Two appendixes help the reader: Logic and Rules of Inference and Symbolic Computation

The part on syntactic processing begins with an introduction to linguistic theories of syntax. It discusses top-down, bottom-up, and mixed-mode parsing strategies, including transition networks and chart parsers These schemes are extended in the chapter on features and augmented grammars. Movement and hold lists are considered next, with a brief digression on slashed categories This part ends with a discussion of deterministic parsing, concentrating on Marcus-style parsing with lookahead buffers

The part on semantic interpretation starts with a chapter explaining logical form. Logical form is Allen's intermediate semantic formalism and is a version of case frames It is extensively used throughout the remainder of the book. Several semantic interpretation strategies are then described, ranging from syntax-first systems to parallel syntactic-semantic analysis. The final chapter on semantics briefly introduces problems such as scoping, noun modification, word sense, and tense and aspect

Part 3 discusses context and world knowledge, starting with a description of logic- and frame-based representation systems. Partial solutions to problems of reference resolution are given using history lists A discussion of scripts and plans follows, with a section on plan recognition Some methods for analyzing discourse

structure are presented, involving the use of cue phrases, tense, reference, and plan recognition to break a sequence of sentences into segments. A chapter on belief models and speech acts concludes this part.

Part 4, Response Generation, makes the book title something of a misnomer. This part seems to cover not only generation but also other topics that did not fit into earlier sections of the book For example, even though the bulk of LUNAR deals with natural language understanding, LUNAR is included under the discussion of response generation, probably because its procedural meaning representation is incompatible with the previous parts. Some real generation issues are covered as well, especially the asymmetries between understanding and generation.

A valuable feature of this book is that it relates many different schemes. This feature is achieved implicitly. consistent use is made of one notational format—Allen's own logical form plus fairly generic versions of a lexicon, context-free grammar, and semantic network–type hierarchy. Formalists might complain that no formal semantics are given for Allen's notation, but the surrounding discussion gives ample clues for the concerned reader to construct a logic-based semantics. Allen then takes various authors' work and transforms it into his notation. Thus, readers acquire a better feeling for the similarities between schemes than if they read the original works.

The original sources for the material are credited in the Related Work sections that close each chapter, and occasional divergences between the original work and this presentation are explained. The Related Work sections are some of the book's best reading; for example, in the passage on lexical functional grammar (p 118), Allen makes a good start toward relating his notation to the standard. However, the sections could be expanded, especially with regard to other standard formalisms, such as KL-ONE Such information is vital if the book is to be used as a reference. Discussing other formalisms would also allow differences between schemes to be made more explicit; as it is now, it is often difficult to perceive why and when one scheme is preferable to another

In fact, if there is one prevailing weakness, it is the shortage of discussion on the goals and theoretic strengths and weaknesses of the various architectures The algorithms are presented one after the other without enough motivation to help the reader evaluate one scheme over another. For example:

> To handle ellipsis the system needs to maintain the complete syntactic analysis of the last sentence (or two sentences, in the case of a dialogue). To parse the input fragment the system needs to maintain a chartlike structure while parsing (p. 359).

It is hard to discern whether this statement describes a particular system's implementation details or whether it makes a theoretic claim for the necessity or sufficiency of a two-sentence dialogue memory. Most of the time, it seems Allen only intends to suggest there might be theoretic points, but as a consequence, the reader is left alone to extract the theory from the algorithms.

The outstanding exception is the part dealing with semantic interpretation, where Allen progresses from one architecture to the next, from a syntax-first system to a parallel architecture, using the same test bed, with side-by-side charts comparing the intermediate processing stages of each scheme The advantages of parallel processing are immediately obvious, namely, the reduction in number of unnecessary parses. From these chapters, the reader acquires the necessary sense of perspective to compare architectures, and these chapters alone make this book a worthwhile acquisition.

It could be that Allen's reluctance to make theoretic claims arises from a desire to maintain objectivity in a general text for a controversial field Still, it would be better to use explanations such as "proponents of augmented widget parsing argue that it handles generalizations about ZZZ-movement more cleanly than ATNs" rather than leave out the explanations altogether. Including the definitions would clarify situations where the goals of linguists, AI researchers, and cognitive scientists legitimately conflict. As it is, the mingling of details with principles sometimes obscures why Allen pursues the goals he does. It is not clear in the discussion of ATN parsing, for example, why a natural language understanding system should rule out all ungrammatical sentences. Besides merely advising the reader how to eliminate ungrammatical parses, it would be better to explain the circumstances where ungrammatical utterances should be accepted (for example, for robustness) as well as the theoretic linguistic concerns underlying notions of ungrammaticality. Currently, one unfortunate tendency by the naive reader is to swallow the whole scheme unaware—hook, line, and sinker. Allen should take advantage of the book's comprehensiveness; it is a terrific opportunity to explain how the theoretic foundations of various schemes relate.

The book's major unspoken assumption is its orientation toward centralized control flow. The coverage of spreading activation techniques is brief (three pages), and only one Related Work paragraph contains references to several well-known connectionist natural language models. Although full coverage of these areas is arguably beyond the book's scope, they are significant and different enough to warrant an explanation of the consequences of distributed control flow. The section on spreading activation describes it as a technique to compute semantic distance in a type hierarchy but should more clearly distinguish a centralized architecture that employs a spreading activation module from an architecture in which spreading activation dictates the entire flow of control (and, thereby, search order). Despite my earlier contention that a text in a developing field such as natural language processing cannot cover everything, Allen's characterization of connectionist natural language processing as an "area attracting considerable attention" seems to deserve some exposition Perhaps the lack of coverage is due to the recency of much of the groundwork for connectionist natural language processing.

This book is better suited as a textbook than as a reference. For effective use as a course text, the instructor should be prepared to cover the material from the beginning rather than selecting certain chapters One reason is the earlier-mentioned lack of an explicit theoretic framework The advantage of cutting away the theoretic overhead is that Allen is able to streamline the progression of ideas, relying on consistent use of his own notation to unify diverse ideas. This approach makes sense for an introductory text because it reduces excess sources of confusion, but it makes the book less usable as a reference. Another reason it makes a better text than reference is that Allen extends his notation incrementally, from section to section, throughout much of the book. The progression is logical, but no centralized definition exists for his notation for easy lookup. The advanced reader who begins in the middle will likely through earlier chapters in search of buried explanations of minor notational features.

Additional organizational quirks arise from the book's orientation as a textbook. For reference purposes, the book's placement of boxes is questionable, although the placement is largely acceptable for teaching purposes. For example, the notion of presuppositions is introduced in Box 16.2 in the part on response generation under question-answering systems. Although mildly related to the surrounding text (natural language interfaces for database retrieval), one would expect to find presupposition in the earlier chapters on reference, discourse structure, or belief models.

The exposition is dry but reasonably concise, with the exception of the English descriptions of the book's algorithms and data structures. The frequent step-by-step descriptions and traces of algorithms are sometimes so detailed that one would almost prefer Lisp code Tedious though it can be, the detail serves its purpose, which is to enable even those with weak computational backgrounds to understand the algorithms Moreover, the constant use of concrete examples is a big plus.

Minor gripes include such items as the lack of analysis of any natural language other than English and the lack of discussion about difficulties involved in memory-intensive types of interpretation, such as for noun compounds.

The book's orientation differs from the few other introductory-level books that cover a wide spectrum of natural language processing issues. Cullingford's *Natural Language Processing* (Totowa, N.J : Rowman and Littlefield, 1986) is more a description of a particular dialogue system than a treatment of many different techniques; this text covers in much greater depth the advanced issues, such as pronominal reference resolution. Winograd's *Language as a Cognitive Process Volume 1—Syntax* (Reading, Mass.: Addison-Wesley, 1983) is more comprehensive in its coverage of syntactic techniques more comprehensively than this text (and gives good motivational discussion as well); unfortunately, volume 2, dealing with semantics and pragmatics, has apparently been shelved Alternatively, the concise natural language processing sections in Charniak and McDermott's *Introduction to Artificial Intelligence* (Reading, Mass.: Addison-Wesley, 1985) contain distilled versions of most subjects covered here and would be suitable for use in a general AI course.

This text will appeal to several audiences. It is a good book for an introductory natural language processing course, although the instructor will have to supply students with additional theoretic motivation. For researchers, it provides a quick route to understanding and relating a larger range of work in computational linguistics at the basic level. The developer seeking to build a natural language system using working technology will also find many useful algorithmic and architectural descriptions. Despite some shortfalls, this book combines impressive breadth and depth in a single volume.

Dekai Wu is a post-graduate researcher at the Computer Science Dept , 545 Evans Hall, UC Berkeley, Berkeley, CA 94720