American Journal of Computational Linguistics *Microfiche 72*

THE FINITE STRING

NEWSLETTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

VOLUME 14 - NUMBER 7 DECEMBER 1977

RELEASED FOR PUBLICATION ON MARCH 3, 1978.

AMERICAN JOURNAL OF COMPUTATIONAL LINGUISTICS is published by the Association for Computational Linguistics.

COMPUTATIONAL SEMANTICS

AN INTRODUCTION TO ARTIFICIAL INTELLIGENCE AND NATURAL LANGUAGE COMPREHENSION

EDITED BY EUGENE CHARNIAK AND YORICK WILKS

Institute for Semantic and Cognitive Studies
University of Geneva

University of Edinburgh
Now: University of Essex
Colchester, Essex, England

NORTH-HOLLAND PUBLISHING COMPANY

P. O. Box 211
Amsterdam, The Netherlands

52 Vanderbilt Avenue
New York, New York 10017

xiii + 294 pages LC 76-10133

$19.00 (Dfl 55) ISBN 0-7204-0469-X

REVIEWED BY STUART C. SHAPIRO
Computer Science Department
State University of New York
Buffalo 14214

Introduction

This is the first book I know of that is devoted to teaching about computer understanding of natural language other than research monographs or anthologies of research papers. It derives from the lecture notes for a turorial conference given at the Institute for Semantic and Cognitive Studies in Switzerland in 1975. The theme is stated at the outset,

> "Computational Semantics, the name we have given here to the study of language based upon Artificial Intelligence methods, therefore approaches language by asking how language is used in translating languages, question answering based on language texts, etc. This approach assumes that 'language is as language does , an idea not unknown to the older disciplines of linguistics, psychology, etc., but Computational Semantics is unique in making this idea the basis for the entire study of language. It is the contention of this approach that it is at best ill advised, and at worst meaningless, to talk of 'understanding' without reference to some task in which language is being used, whether as narrow as sentence completion questions in an IQ test, or as vague as flirting at a party" [p.1]

The twelve chapters are written by seven different authors. Below, each chapter is discussed separately, after which some overall comments are given.

Inference and Knowledge Part I by Eugene Charniak (21 pages)

In this chapter, Charniak discusses first order predicate calculus (FOPC), and the programming language PLANNER.

His discussion of FOPC includes brief informal introductions to the usual syntax of classical FOPC and to the resolution principle. It is too brief and too informal for the reader to come to any understanding of the subject and certainly inadequate to enable the reader to form an opinion on the usefulness of FOPC or resolution. Yet Charniak states his opinion, including the patently false notions that "the general idea behind FOPC is that one only makes inferences when one is asked a question... When you come right down to it, FOPC is primarily a theory of inference mechanism". [p.9] As Hayes [11] points out, logic provides an analysis of meaning. It is not an inference mechanism, rather "Logical meaning justifies inferences", [p.559, italics in original]. Resolution is not even an inference machanism. It is not correct to say, as Charniak does, that "resolution has only one rule of inference" [p.8]. Resolution is a rule of inference as are Modus Ponens and Substitution. Many different mechanisms have been studied for applying resolution, several of which are discussed in [3]. There are even FOPCs which are radically different from the classical one that Charniak discusses. For example, in intuitionistic FOPC $A \vee \sim A$ is not a theorem (see [13]), and, contrary to Charniak's statement that "it is a well known property of FOPC that anything can be proved from a contradiction", in entailment logics [1] $(A \& \sim A) \supset B$ is not a theorem. It is important for computational semanticists to realize that such logics exist (see [19]). It is also important to realize, as Charniak apparantly does not, that the implementation of *any* inference machanism entails an underlying logic. The implementor should have some idea of what that logic is. Is it consistent? If not, what is to be done when inconsistencies become apparant? How complete is it? Is there any easily identifiable class of questions that cannot be answered?

Charniak discusses the useful distinctions of inference at "question time" versus at "read time" and "problem occasioned" versus "non-problem occasioned" inference, but goes awry by using this discussion to fault FOPC, which he again calls "a question driven system for making inferences". [p.13]

Charniak briefly describes Raphael's SIR and then discusses PLANNER. This discussion is also brief, but well written and should give the reader a good introduction to the basic ideas of PLANNER. The final comparison of

PLANNER with FOPC is, of course, misguided since PLANNER is a programming language while FOPC is not.

One feature of PLANNER that Charniak includes in his examples, but does not discuss as much as it deserves is THNOT. (THNOT A) succeeds if A is not in the data base and cannot be deduced by the invoked theorems. This is quite different from A, which is <u>true</u> if A is <u>false</u>. This brings up the interesting point that most question answering systems assume a three valued logic (<u>true</u>, <u>false</u>, <u>neither</u>) rather than the classical two valued logic (note that logicians have studied multiple valued logics and that intuitionistic logic admits a status other than A or A). It also makes one wonder what to do if after using (THNOT A) to deduce B, A is put into the data base, a concern Charniak does not discuss. Another incomplete discussion concerns what Charniak calls "coping with contradiction" and Raphael has called the exception principle. [15] The example problem is "all people have two legs" but "Bill has one leg". Charniak says that "in PLANNER this is done by making 'Bill has one leg' an assertion, and 'All people have two legs' a theorem. Since the data base is checked first when trying to establish a goal, the system will find that Bill has one leg before it attempts to use the general theorem to show that he has two legs" [p.21] This is not coping with contradiction, it is ignoring it. What if we then ask "Who has two legs?" and learn via the theorem that Bill does? Also, if we have the theorems "All birds fly" and "All penguins don't fly", and the assertions "Billy is a penguin" and "Billy is a bird", how are we to know what the system will deduce about Billy's flying? We need a tighter logical understanding of our inference mechanisms than Charniak seems to think.

This reviewer's final conclusion about this chapter is that it is a good introduction to several topics, but it dangerously contributes to a misunderstanding of logic among its intended audience. In the second chapter, Charniak says, "AI programs are going to need knowledge of syntax, anyway, so why not use sources at hand". [p.39] This chapter should have the statement, "AI programmers are going to need knowledge of logic, anyway, so why not use sources at hand".

Syntax in Linguistics by Eugene Charniak (18 pages)

This chapter covers "syntax within the theory of transformational grammar" as of approximately 1965, the era of Chomsky's Aspects of the Theory of Syntax [5]. It is a good introduction to the topic, including sections on, "The Nature of Linguistic Arguments", "Some Typical Transformations", "What is a Transformational Grammar", and "Relation of Transformational Grammar to Artificial Intelligence", in which Charniak laudably argues that "both disciplines have something to say to each other" [p.36] while repeating one of the themes of the book, that "what artificial intelligence does offer is the opportunity to attack the real fundamental problem, language comprehension, without worrying so much about sentence grammaticality". [p.40]

Semantic Markers and Selectional Restrictions by Philip Hayes (14 pages)

In this section, Hayes discusses the influential semantic theory of Katz and Fodor [12] and the objections raised to it by Dwight Bolinger [2]. He ties this to the general topic with a section on the "Use of Semantic Markers and Selectional Restrictions in AI". Once again, the chapter is brief, but well written and serves to introduce the topics.

Case Grammar by Wolfgang Samlowski (18 pages)

Samlowski begins by discussing Fillmore's theory of case grammar as presented in his "Case for Case" [10]. He then discusses the case systems used by three A.I. workers Simmons, Schank and Wilks - and compares their cases to Fillmore's and to each other's. This chapter is a good introduction to the topic.

Generative Semantics by Margaret King (16 pages)

"This chapter falls into two sections. The first section attempts to display the Generative Semantics position, by reporting, without much criticism or comment, arguments which are believed by their authors to justify that position... In the second section some aspects of the Generative Semantics position will be considered more critically" [p.73] The main criticism is that "the attempt to include presupposition, in a very broad sense, in a grammatical system leads to a situation where grammar breaks down". [p.86] King's basic conclusion regarding Generative Semantics is that, "while it is basically misconceived as an activity within traditional linguistics, [it] would nonetheless be a perfectly sensible activity within the general area of work in AI". [p.73]

Whether or not one would want to argue with Section II, Section I gives a good summary of the arguments of McCawley, Lakoff and others deriving the Generative Semantics position from Chomsky's "Aspects" model.

Parsing English I by Yorick Wilks (12 pages)

This chapter only minimally fulfills the promise of its title, probably due to Wilks' conviction that "grammatical (or syntactic) parsing of the sort described is not fundamental, and that it need not be even a preliminary to assigning a useful meaning structure to sentences". [p.92] Wilks presents a six rule, four lexeme, non-recursive, context free grammar and uses it to do a top-down and a bottom-up parse of "The dog likes the cat" (one of the four sentences in the language of the grammar). He also mentions lexical ambiguity and breadth-first versus depth-first parsing.

Wilks spends about half the chapter discussing Winograd's SHRDLU. While this may be appropriate, Wilks, because of his bias, fuzzes the distinction between Winograd's parser ("The parser is an interpreter which accepts recognition grammars written in a procedural form. The formalism is a language called PROGRAMMER". [20,p.3]) and his blocks world robot system. Wilks claims that, "Indeed, it might be argued that, in a sense, and as regards its semantics, Winograd's system is not about natural language at all, but about the other technical questions of how goals and sub-goals are to be organized in a problem-solving system capable of manipulating simple physical objects". [p.99, italics in original] While that may be true about the robot system, Winograd's own distinction between parsing and problem solving ("Even though we used the robot system as our test area, the language programs do not depend on any special subject matter, and they have been adapted to other uses". [20,p.2]) should not be ignored, especially in a chapter on parsing. The same mistake is made with respect to Woods' Augmented Transition Network (ATN) Grammars. While it is true that, "both Woods and Winograd have argued in print that their two systems are essentially equivalent" [p.99, italics in original], and this equivalence is accepted by the AI community, the systems that are equivalent are the parsing systems not the robot system and the lunar rocks question-answering system [21]. Thus it is quite wrong, relative to what should be discussed in this chapter, to say that "both are grammar-based deductive systems, operating within a question-answering environment in a highly limited domain of discourse". [p.99] It is also incorrect to say about the parsers that "there is no need to discuss both, and Winograd's is, within the AI community at least, the better known of the two". [p.99] Indeed, the large majority of AI language understanding systems use ATN grammars, and the absence of a discussion of them is one of the greatest shortcomings of this book.

Semantic Nets as Memory Models by Greg Scragg (27 pages)

In this chapter, Scragg introduces and discusses semantic networks from those with arc labels such as LIKES, HIT and HAP (has-as-part), which should probably just be termed "relational" networks, to those with case relations as arcs, to discussions of quantification and of procedures in semantic networks. He also takes a few pages each to discuss Schank's and Simmons' networks.

Scragg rightly discusses the difference between individuals and classes (he uses the terms "tokens" and "types"), and the importance of distinguishing set membership from subset, a surprisingly often neglected and confused point. However, he continues a closely related confusion by using the same relation, HAP, between tokens (a token of GIRL and a token of HAIR) as between types (the types BIRD and WING). This is incorrect because the interpretation of x HAP y cannot be consistent. In some cases, it is "x has y as part", and in others it must be "each x has a (distinct) y as part".

Scragg gives some insight into the data structures for implementing semantic networks and so encourages the reader to look beyond the usual pictorial representation. This is important when comparing different network formalisms. For example, as Scragg points out, Schank's pictures look radically different from any other semantic network pictures, yet the actual data structures are very similar.

<u>Inference and Knowledge Part 2</u> by Eugene Charniak (26 pages)

Charniak begins this chapter by discussing the demon based system of [4]. He criticizes this approach and also PLANNER (demons are PLANNER antecedent theorems) because of the fixed direction of excitation. For example, if the possibility of rain activates the umbrella demon, how do we understand "Jack began to worry when he realized that everyone on the street was carrying an umbrella" [p.136]? The basic problem seems to be that PLANNER requires one to distinguish one of the propositions of an inference rule as a pattern, burying the others inside the theorem. It does not allow all of the propositions to be treated as patterns, as necessary (as is allowed by the semantic network deduction rules of [18]).

Next Charniak discusses McDermott's TOPLE [14]. The interesting features discussed are TOPLE's sets of possible worlds and its performing inferences in order to supply support for believing new inputs. He also discusses Rieger's inference program [16], concentrating on Rieger's belief in massive read time inferencing and his classification of sixteen types of inferences.

Finally, Charniak discusses the influential, though controversial theory of frames. He compares frames to demons and finds frames preferable.

<u>Parsing English II</u> by Yorick Wilks (30 pages)

This chapter is a continuation of Parsing English I. Apparently, they were originally written as one chapter, then separated for no obvious reason. In this chapter, Wilks warns that, "'parsing' is being used not only in its standard sense in mathematical and computational linguistics" [p.155], but includes building some meaning structure representation of the surface language. This kind of parsing Wilks definitely favors: "The thesis behind this chapter ... is that parsing is essential to a system... The argument is not only that parsing provides a test of a proposed structure, for that is secondary, but that the parsing procedures *define what the significance of the proposed structure is*" [p.179, italics in original].

Wilks first discusses three "second generation" parsing systems. The keyword and pattern parser of Colby et al.'s PARRY [6;7;9], Wilks' own preferential semantics system based on "formulas", "templates" "paraplates" and "inference rules", and Riesbeck's parser for the MARGIE system [17]. About ten pages are spent on Wilks' system, more than any other system discussed in the book.

Wilks considers all the systems discussed in this chapter to use "frames". He feels, "the key point about any structures that are to be called frame-like is that they attempt to specify in advance what is going to be said, and how the world encountered is going to be organized... In psychological and visual terms, frame approaches envisage an understander as at least as much a looker as a seer" [p.156]. He distinguishes between "small scale" and "large scale" frames (large scale frames are what are commonly referred to as frames or "scripts"), and has some scepticism about large scale frames, "It is not being argued here that large scale frames have no function, only that, as regards concrete problems of language understanding, their function has not yet been made explicit" [p 183].

The chapter ends with a nine page comparison of systems on ten different dimensions: level of representation; centrality of information; the phenomenological level of inferences; decoupling of parsing and inference; exhibition of manner of application to input texts; amount of forward inferencing; odularity; scale of representation; connection with real world procedures; justification of adequacy.

<u>Psychology of Language and Memory</u> by Walter F. Bischof (19 pages)

"The intent of this chapter is twofold: first it should provide the non-psychologist with some basic concepts and some important experimental findings in the field of psycholinguistics and the psychology of memory. The second goal is to take a close look at the nature of psychological arguments and psychological evidence" [p.185]. This is done from an admitedly biased point of view: "the topics chosen for review were chosen more because of their popularity in AI than because of their relative importance in psychology" [194]. The topics include association theory, experiments designed to test the psychological reality of phrase-structure and transformational rules, memory of meaning versus memory of syntax, short term and long term memory, the Collins and Quillian model [8] of hierarchical memory organization. Although the discussions are brief, the choices are good for the intended audience.

Bischof is not only selective in his view of psychology, but highly skeptical: "the student of AI should be able to see from the discussion here that the ability of psychology to design and carry out experiments which will give clear and indisputable results is very limited, and that their ability to provide a safe and clear insight into human language understanding is similarly limited" [p.194]. Therefore, "AI is well advised not to over-estimate the importance of psychological arguments" [p.201].

Philosophy of Language by Yorick Wilks (29 pages)

In this chapter, Wilks is mainly interested in discussing and comparing the work of Richard Montague and Ludwig Wittgenstein. "These philosophers have been chosen not so much for their influence on our subject matter, which has been small, but because their views are diametrically opposed on the key issue of formalization" [p.205]. His belief is "that the influence of Wittgenstein has been largely beneficial while that of Montaque has been largely malign" [p.205].

First Wilks introduces some basic topics from the work of Leibniz, Frege, Russell, early Wittgenstein, Carnap, and Tarski. As in the previous chapter, the discussions are brief but the topics well chosen. His introduction to Montague is via the analysis of the sentence "Every man loves some woman". Montague is always difficult, but Wilks' presentation is relatively easy to follow and is done without introducing the lambda calculus. His discussion of Wittgenstein is intended, "simply to give a flavor, to those unfamiliar with him, of what Wittgenstein has to offer" [p.222] His style is to cover eight topics by presenting for each a thesis, some quotes from Wittgenstein and some comments. The overall impression is that Wilks feels that Wittgenstein is relevant to AI workers and that Wittgenstein basically supports the view of language understanding put forth by this book.

<u>An Introduction to Programming in LISP</u> by Margaret King and Philip Hayes (47 pages)

This chapter is precisely what its title suggests - an introduction to LISP for either the non-programmer or the non-LISPing programmer. It takes a modern approach (dotted pairs are not mentioned) and uses examples to which the book's audience will be able to relate. It also has a good set of exercises at the end of each section with solutions at the end of the chapter. Of necessity it moves fast, skimming through many topics including PROGs, property lists, mapping functions and FEXPRs. This sheds doubt on how much of a LISPer the naive reader will become after working through the chapter. Like much of the book, all the right topics are covered; but briefly.

Conclusions

If one were to design a course on natural language understanding by computer, and list all the topics that should be covered, one would find that almost all were included in this book, ATN grammars being one notable exception. However they are al covered only briefly and from a very definite point of view. I have just finished giving such a course using this book, additional readings, and my own "corrective" viewpoint. The students felt that the book's discussion of each topic was too brief to be self-contained unless they already knew something about it.

References

1. Anderson, A.R., and Belnap, N.D., Jr. Entailment: The Logic of Relevance and Necessity, Princeton University Press, Princeton, NJ, 1975.

2. Bolinger, D. The atomization of meaning, Language 41, (1965), 555-573.

3. Chang, C.-L., and Lee, R.C.T. Symbolic Logic and Mechanical Theorem Proving, Academic Press, New York, 1973.

4. Charniak, E. Toward a model of children's story comprehension. TR-266, MIT A.I. Lab., Cambridge, MA, 1972.

5. Chomsky, N. Aspects of the Theory of Syntax, MIT Press, Cambridge, MA, 1965.

6. Colby, K.M., et al. Artificial paranoia. Artificial Intelligence 2, (1971), 1-25.

7. Colby, K.M., and Parkinson, R.C. Pattern-matching rules for the recognition of natural language dialogue expressions. AJCL Microfiche 5, 1974.

8. Collins, A.M., and Quillian, M.R. Retrieval time from semantic memory. J. Verbal Learning and Verbal Behavior 8, (1969), 240-247.

9. Enea, H., and Colby, K.M. Idiolectic language analysis for understanding doctor-patient dialogues. Proc. Third

IJCAI, SRI, Menlo Park, CA, 1973.

10. Fillmore, C.J. The case for case. In E. Bach and R.T. Harms (Eds.), Universals in Linguistic Theory, Holt, Rinehart and Winston, New York, 1968.

11. Hayes, P.J. In defence of logic, Proc. IJCAI-77, MIT, Cambridge MA, 1977, 559-565.

12. Katz, J., and Fodor, J.A. The structure of a semantic theory, Language 39, (1963), 170-210. Also in J.A. Fodor and J.J. Katz (Eds.) The Structure of Language, Prentice-Hall, Englewood Cliffs, NJ, 1964, 479-518.

13. Kleene, S.C. Introduction to Metamathematics, D. Van Nostrand Co., Princeton, NJ, 1950.

14. McDermott, D.V. Assimilation of new information by a natural language understanding system. TR-291, MIT A.I. Lab., Cambridge, MA, 1974.

15. Raphael, B. SIR: a computer program for semantic information retrieval. In M. Minsky (Ed.). Semantic Information Processing, MIT Press, Cambridge, MA, 33-145.

16. Reiger, C. Conceptual Memory. Unpublished Ph.D. Thesis, Stanford U., 1974.

17. Riesbeck, C.K. Conceptual analysis. In R. Schank. Conceptual Information Processing, North-Holland, Amsterdam, 1975, 83-156.

18. Shapiro, S.C. Representing and locating deduction rules in a semantic network. Proc Workshop on Pattern-Directed Inference Systems, SIGART Newsletter, 63 (June, 1977), 14-18.

19. Shapiro, S.C., and Wand, M. The relevance of relevance, Technical Report No. 46, Computer Science Dept., Indiana U., Bloomington, IN, 1976.

20. Winograd, T. Understanding Natural Language. Academic Press, New York, 1972.

21. Woods, W.A.; Kaplan, R.M.; Nash-Webber, B. The lunar sciences natural language information system: final report, Report 2378, Bolt Beranek and Newman, Cambridge, MA, 1972.

INTRODUCTION TO CONTEMPORARY LINGUISTIC SEMANTICS

GEORGE L. DILLON

Indiana University -- Purdue University
Fort Wayne

PRENTICE-HALL, INC.
Englewood Cliffs, New Jersey 07632

xx + 150 pages 1977 LC 76-41853
$6 95 paper ISBN 0-13-479469-9

REVIEWED BY JAMES D. MCCAWLEY

University of Chicago

This book reminds me of the 1969 Chicago Cubs, who held a commanding lead in the National League until early September but went into a tailspin and finished far behind the New York Mets. The first five chapters of the book, in which Dillon goes through a lot of lexical semantics, introduces many basic notions of semantics, and discusses the relationship of semantics to morphology, to extralinguistic knowledge, and to metaphor, are excellent: he gives a clear, coherent, and accurate presentation of a well-selected and quite broad-ranging series of topics. However, the sixth chapter, dealing with formal logic, is so thoroughly bungled that I find it hard to believe that it could have been written by the same person that produced the preceding chapters.

I should mention at the outset that I have an indirect personal connection with this book, namely that I was the first person (Dillon is the third) who was engaged to write it; I withdrew from the project a year after my deadline, with only 17 pages of manuscript written. Dillon's book is in fact not all that different from what I might have written if I had been able to get my act together in about 1970. (Dillon, I should point out, did not see my aborted manuscript, not that it would have done him any good). Indeed, many of the bungles in chapter 6 for which I chide Dillon can be attested in my own works of the late 1960's, which Dillon drew on in preparing chapter 6. The fact that I will devote more space here to saying what is wrong with chapter 6 than what is good about chapters 1-5 and 7 should not be taken as implying that I think the faults of the one bad chapter outweigh the virtues of the other six. In fact, the book could be used quite profitably in a course on semantics or in an introductory linguistic course, provided that the instructor either has his students skip chapter 6 entirely or has them read a good deal of supplementary material on the topics covered there.

The principal failings of chapter 6 are a failure to separate the many distinct questions that arise, systematic errors in the employment of standard logical formalisms, bizarre English paraphrases of logical formulas, and implausible claims about the meanings of the English sentences used as examples. Dillon appears to accept the policy known as 'unrestricted quantification', in which all variables range over the entire universe of discourse, and the noun of the quantified NP is fit into the logical structure by using a connective ($\supset$ in the case of the universal quantifier, & in the case of the existential quantifier) to combine it with the 'matrix' propositional function, e.g.

(1) a. $(\forall x)$(Philosopher x $\supset$ Dangerous x)

'All philosophers are dangerous'

b. $(\exists x)$(Linguist x & Obnoxious x)

'Some linguists are obnoxious'

This policy is to be contrasted with an alternative not considered by Dillon, the so-called 'restricted quantification', in which each variable ranges over a restricted domain, given by the noun of the quantified NP, with a logical constituent structure as in (2):

(2) a. $(\forall x$: Philosopher x)(Dangerous x)

b. $(\exists x$: Linguist x)(Obnoxious x)

See McCawley 1972 for arguments in favor of restricted quantification as part of a system of logic that can optimally be integrated with natural language syntax and linguistic semantics. For one thing, unrestricted quantifiers are hopeless as a basis for the analysis of quantifiers other than the logicians' favorite ones: any viable analysis of <u>most</u>, <u>many</u>, <u>few</u>, or <u>almost</u>

all would have to involve some version of restricted quantification.

When Dillon says (p. 95) "Generic sentences, though the quantifier is less than universal, have also been symbolized with an entailment"[1], he conflates the issue of how genericity is related to quantification and the issue of how quantifiers fit into logical structure. He ought to have discussed the latter issue long before he took up generics, so that he could deal with the matters that are peculiar to generics against a background of clear alternatives for the treatment of those things to which generics might be related. His elevation of an ancillary issue to central status here is made particularly glaring by the fact that his brief discussion of generics comes in his section on "connectives" rather than the one on quantifiers.

Dillon would have been wiser to do chapter 6 in terms of restricted quantifiers rather than unrestricted quantifiers, since he would then have been able to avoid mechanical difficulties that are inherent in the use of unrestricted quantifiers and which he himself has not mastered. Specifically, in representing the meanings of complex sentences, Dillon never gets the material corresponding to the nouns in the right place. For example, in representing the meaning of (3a) he gives the nearly tautologous (3b) instead of the standard (and plausible) (3c):

(3) a. Each boy kissed a girl.

b. $(\forall x)(\exists y)((BOY(x)\ \&\ GIRL(y) \rightarrow KISS(x,y)))$

c. $(\forall x)(BOY(x) \rightarrow (\exists y)(GIRL(y)\ \&\ KISS(x,y)))$

According to the standard truth conditions, (3b) is true under virtually all circumstances, namely any circumstances under which there are entities that are not girls (the number 10 is not a girl, therefore (BOY(x) & GIRL(10)) is false whatever x is, whether a boy or a beanbag, and thus for any value of x there is a value of y, namely 10, that makes the conditional in (3b) true). The fact that Dillon allowed formulas like (3b) to appear in his book can be attributed in part to his lack of attention to the giving of rules for the relationship between logical structure and surface structure. The most obvious rules for mapping unrestricted quantifier formulas onto surface structures would cover the case of (3c) but not that of (3b), since (3c) but not (3b) consists of structures like (1a-b) embedded in one another. The very first example that he gives of an analysis involving quantifiers is an analysis of (4a) as (4b) rather than as (4c):

(4) a. A boy kissed a girl.

b. $(\exists x)(\exists y)((BOY(x)\ \&\ GIRL(y))\ \&\ KISS(x,y)))$

c. $(\exists x)(BOY(x)\ \&\ (\exists y)(GIRL(y)\ \&\ KISS(x,y))$

While (4b) and (4c) have exactly the same truth conditions, it is (4c) that fits the general rules associating unrestricted quantifier formulas with surface structures containing quantified

NP's. Had his discussion of (4a) been preceded by the material that would need to be covered to make that fact obvious, Dillon could scarcely have made such blunders as (3b).

One recurring disturbing feature of Dillon's informal glosses to his logical formulas is his highly unidiomatic use of the word <u>one</u>, as when he glosses (4b) as 'There exists one that is a boy and there exists one that is a girl such that he kissed her'.[2] This same odd locution also occurs in earlier chapters, as in his discussion of the semantics of Adjective + Noun combinations (p. 62). where he glosses <u>large car</u> as 'one that is a car with size greater than average size of cars' and <u>joint undertaking</u> as 'one that is an action that is undertaken jointly'. In his Adjective + Noun glosses, the <u>one</u> is in fact superfluous (e.g. one could omit 'one that is from the last two glosses and from the other glosses in that section); however, the <u>one</u>'s would presumably reappear if Dillon were to employ consistently the analysis of relative clauses as derived from coordinate structures that figures in chapter 6 and were to stick to the style of glosses that he uses for (4b). He in fact introduces this odd use of <u>one</u> along with his very first example of a logical formula, namely an analysis of (5a) as (5b), for which he gives the gloss (5c):

(5) a. John kissed Mary.

b. ((JOHN(x)) & (MARY(y)) & (KISSED(x,y)))

c. One is called John and one is called Mary and he kissed her.

If the *one*'s are interpreted literally, (5c) expresses a rather bizarre proposition that has no relationship to (5a). A significant improvement could be obtained by replacing both occurences of *one* by *someone*, but then the gloss would correspond to a formula involving two existential quantifiers, and Dillon makes clear later that he does not want to include any quantifiers in the analysis of (5a). In fact, Dillon's use of *one* is only an obscure way of repeating the very thing that it is supposed to 'explain', namely the index *x* or *y*, and his glosses would have been much clearer if he had simply written *x*'s and *y*'s in them.

Dillon's discussion of (5b) is unclear and/or inconsistent as to whether the *x* and *y* are constants or variables. The absence of a quantifier and the contrast that he draws (p. 87) between (5b) and structures involving quantifiers suggest that they are constants. However, his tabulation of "seven distinguishable situations" that the negation of (5b) is compatible with consists of glosses that would make little sense if *x* and *y* were constants, e.g. "One kissed one called Mary, but he wasn't called John": from the fact that Larry kissed Mary and Larry isn't called John, it does not follow that John didn't kiss Mary (perhaps they both kissed Mary). The different situations that he is distinguishing really involve a third index, one corresponding to the event of kissing. Some of the seven situations in the list involve a presupposition that an act of kissing took place and give specifications of who the participants in <u>that</u> act were; others do not involve such a presupposition. There is in fact no proposition of which the seven situations can all be taken as illustrating the negation.

I turn now to chapters 1-5 and 7, which I rate in general as very well done. In these chapters Dillon discusses componential analysis, productive and non-productive word-formation, and metaphor in terms of a large number of well-chosen examples. I find only one really major gap in the set of topics covered, namely presupposition. While the term 'presupposition' occurs in several places in the text, Dillon contents himself with merely advising the reader that the term has been used in a number of different senses and is involved in ongoing controversies, and refers the reader to supplementary readings for further enlightenment. The various notions to which that term has been applied are of sufficient importance in linguistic semantics and are of relevance to so many of the examples and theoretical points that he takes up that Dillon can hardly expect instructors using his book to avoid serious discussion of it.

Dillon's discussion of metaphor is distinguished by its use of a large body of interesting examples and its avoidance of the hackneyed examples (such as <u>He danced his did</u>, which is not a metaphor at all) that usually figure so prominently in linguists' discussion of metaphor. However, in this generally enlightened presentation there are two errors for which I wish to take Dillon to task. First, he speaks of metaphor as resting on 'some sort of incompatibility between the usual senses of the word and the context' (p. 39). But if metaphoric uses were restricted to contexts with which a literal use was incompatible, an expression could never be ambiguous as to whether it is to be interpreted literally or metaphorically, whereas in fact such ambiguity is quite common (Reddy 1969). Indeed, as Ted Cohen (lecture at University of Chicago) points out, there are metaphoric sentences that express truisms when interpreted literally (e.g. No man is an island). Secondly, the adjustment involved in interpreting a metaphor is not (as Dillon says it is) the cancellation of semantic features but the assignment of a non-standard referent; for example, I disagree with Dillon's statement that the interpretation of morsel in "Broad-fronted Caesar / When thou wast here about the ground, I was / A morsel for a monarch" (Antony and Cleopatra I.v.29-31) involves "suppressing the SMALL BIT OF FOOD component in favor of the associated DELICACY component" (p. 40): Cleopatra here is speaking of herself as a snack for the emperor, and the reader transfers the reference (though not the sense) of the "EATING" or "CONSUMING" component to a different medium rather than simply suppressing it (see again Reddy 1969).

Useful problems are given at the end of each chapter, with suggested answers given at the end of the book. There is also a glossary that is quite useful though flawed in some respects, chiefly incompleteness, as whe Dillon declines to even hint at a definition of 'presupposition' and simply refers the reader to other literature, and when he defines 'achievement verb' (the term is from Vendler 1957) by telling the reader everything that it isn't and leaving the reader to determine by elimination what it is. The definition of 'connectiv contains a statement that confuses an important issue: "Connectives are held to be predicates by some, but not by logicians, because predicates combine with arguments to form propositions, but connectives combine propositions to form larger propositions" (p. 124). Dillon has given no reason for supposing that propositions can not be arguments of predicates and thus for not taking connectives to be simply a special kind of predicates. Indeed, not to allow propositions to serve as arguments is to commit oneself to the schizophrenic position (Prior 1971) that verbs such as know and believe are "predicates on the left and connectives on the right".

FOOTNOTES

1 Dillon consistently refers to his conditional connective as "entailment" but usually uses it in a way that would only make sense if it were a material conditional rather than entailment. He does not advise the reader of the distinction between the truth-functional material conditional connective that figures in most logic texts and the relation (not really a "connective") of

entailment (A entails B if B is true in all states of affairs in which A is true).

[2] This gloss would in fact be more appropriate for (4c) than for (4b), though the use of one would be no more idiomatic.

REFERENCES

McCawley, James D. 1972. A program for logic. In D. Davidson and G. Harman (eds.), Semantics of natural language (Dordrecht: Reidel), 498-544.

Prior, A. N. 1971. Objects of thought. Oxford: Clarendon.

Reddy, Michael. 1969. Metaphor and reference. Papers from the Fifth Regional Meeting, Chicago Linguistic Society.

Vendler, Zeno. 1957. Verbs and Times. Philosophical Review 64, 143-160. Also in Z. Vendler, Linguistics in Philosophy (Ithaca: Cornell University Press, 1967), 97-121.

CURRENT BIBLIOGRAPHY

The production and editorial staff thanks Martin and Iris Kay, and the Xerox Palo Alto Research Center for their aid in preparing this bibliography.

The LIFER Manual: A Guide to Building Practical Natural Language Interfaces

Gary G. Hendrix
Artificial Intelligence Center, Stanford Research Institute, Menlo Park, CA 94025

Technical Note 138, 68 pp., February 1977

LIFER is a practical system for creating English language interfaces to other computer software and is composed of two main parts: a set of interactive language specification functions and a parser. In standard practice, an interface builder uses the language specification functions to define an application language (an appropriate subset of a NL). Using this language specification, the LIFER parser can translate NL inputs into appropriate interactions with the application software. Topics: A) The LIFER approach to language, B) Specifying a language definition, C) Using the parser, D) The ellipsis feature, E) Spelling correction and other error messages, F) Initial control characters, G) Auxiliary features, H) Implementing pronouns, I) Implementation.

GENERAL

Computer Construction of Crossword Puzzles Using Precedence Relationships

Lawrence J. Mazlack
Computing and Information Science, University of Guelph, Ontario, Canada

Artificial Intelligence 7: 1-19, Spring 1976

After an unsuccessful attempt to construct puzzles by whole insertion of words, puzzles were successfully constructed by a letter by letter method. Usually when a word was validly formed by the letter by letter puzzle constructor it could remain permanently in the constructed puzzle. A dynamic, heuristically determined, decision structure was required. The constructor resolved questions of letter selection, ordering and reordering of the solution sequence, dictionary structure and access, and decision path selection.

GENERAL: PHILOSOPHICAL FOUNDATIONS

Artificial Intelligence, Language, and the Study of Knowledge

Ira Goldstein, and Seymour Papert
Massachusetts Institute of Technology, Cambridge, MA 02139

Cognitive Science 1: 84-123, January 1977

Recent work in AI suggests that intelligence is based on the ability to use large amounts of diverse kinds of knowledge in procedural ways (e.g. frames, scripts), rather than on the possession of a few general and uniform principles (e.g. heuristic search). Within this general framework a fundamental contribution of AI to epistemology is clear: the systematic introduction of active agents into epistemological theory construction so that items of knowledge are active agents. Other contributions of AI include concepts of system self-knowledge (the system's ability to observe its behavior, and to make use of those observations, even to the point of learning to debug faults in its procedures) and the development of a variety of control structures (e.g. ATNs). Finally, the paper considers ways in which AI may have a radical impact on education if the principles which it utilizes to explore the representation and use of knowledge are made available to the student to use in his own learning experiences.

The Past, Present, and Future of Computational Linguistics

David G. Hays
Department of Linguistics, SUNY at Buffalo, Amherst, NY 14260

Papers in Computational Linguistics, Akademial Kiado, Budapest: 583-585, 1977?

The field of computational linguistics is part of the slowly unfolding revolution in man's way of thinking about thought itself, a revolution spurred and supported by the computer. As revolutionaries we have to do without the guidelines that tradition furnishes others; but the luckiest scientists are those like ourselves who, being revolutionaries, have the best chance to make big contributions.

The Field and Scope of Computational Linguistics

David G. Hays
Department of Linguistics, SUNY at Buffalo, Amherst, NY 14260

Papers in Computational Linguistics, Akademiai Kiado, Budapest: 21-25, 1977?

A rich theory of information can be developed only in terms of the classes of machines that are involved in processing it. But a theory of machines can be a competence theory or a performance theory. I propose a four-way schema with psychology, computation, formal linguistics, and descriptive linguistics at the poles. Psychology and computation are about performance; formal sciences are abstract, and psychology and descriptive linguistics are sciences. Psycholinguistics joins psychology with linguistics. Correspondngly, on the abstract side computational linguistics joins computation with formal linguistics and also seems a fruitful area. The most likely place to arrive at a working idea of how competence and performance - algorithms and information - are related is computational linguistics. The more we achieve on the formal, abstract side, the better the chance of formulating goals and criteria for linguistics that will help the linguist decide whether a grammatical invention merits prolonged study.

Computational Linguistics and the Design of Control Systems

David G. Hays
Department of Linguistics, State University of New York at Buffalo, Amherst, NY, 14260

Science and Culture 28: 1426-1432, December 1976 (Sciencia e Cultura)

Five sections: Language, Computation, Computational Linguistics, The Architecture of Correspondence, Control systems. The traditional branches of mathematics analyze structures in which time may play a role, but the analysis does not employ time. Algorithm theory uses time as an analytic variable. Computational linguistics is the algorithmic analysis of language. Control systems operate in time; they receive information about a process as it occurs and return information that influences its continuation. The study of control systems, cybernetics, must therefore, like computational linguistics, use time as an analytic variable. Language is a vehicle for both social and personal control. Computational linguistics must provide the theory with which to understand linguistic control processes. The special qualities of language are needed in many control systems. Computational linguistics now has models for recursive systems in which recursive components must be correlated and may therefore serve usefully as a model for use by the manager dealing with markets within markets and technologies within technologies.

GENERAL: SPEECH UNDERSTANDING

Chapter IV: Experimental Studies

William H. Paxton
Stanford Research Institute, Menlo Park, CA 94025

Speech Understandng Research. Final Technical Report, 15 October 1975 - 14 October 1976, IV-1 to IV-56, October 1976

The first experiment concerned the acoustic processor (the mapper) and the second concerned 'fanout' - the number of alternatives at each word, both for the language alone and in combination with the acoustics. The third experiment studied the effects of four control-strategy design choices. Focus by inhibition and island driving had bad effects, while context checks for priority setting had good effects. Mapping all at once had good effects on everything except acoustic and total runtime, and these bad effects could probably be eliminated by redesign of the mapper. The fourth experiment varied the size of allowed gaps and overlaps between words and showed the potential value of special acoustics tests to verify word-pair junctions. A fifth experiment concerned the effects of increased vocubulary and improved acoustic accuracy while a final study concerned detailed measurements of the Executive performance and provided insights into the use of time and storage and the kinds of errors made by the system.

Chapter I: Introduction

Ann E. Robinson, Donald E. Walker, William H. Paxton, and Jane J. Robinson
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975 - 14 October 1976, I-1 to I-28, October 1976

The system data base contains characteristics such as owner, builder, size, and speed for several hundred ships in U.S., Soviet, and British fleets. The user can get information from the system by simple English questions, commands, and dialog sequences using incomplete sentences and pronouns. After a brief overview (covering the components developed by SDC, the language definition, syntax, semantics, discourse, deduction, generation, and the executive) an example of the system's operation is given and the paper closes with an historical perspective on the system's development.

Speech Understandng Systems: Final Report, November 1974 - October 1976 Volume I: *Introduction and Overview*

William A. Woods, Madeleine Bates, Geoffrey Brown, Bertram C. Bruce, Craig C. Cook, John W. Klovstad, John I. Makhoul, Bonnie L. Nash-Webber, Richard M. Schwartz, Jared J. Wolf, and Victor W. Zue
Bolt Beranek and Newman Inc. 50 Moulton St., Cambridge, MA 02138

Report No. 3438, 82 pp., December 1976

The system arrives at a believable coherent theory that can account for all the stimuli by successive refinement and extension of partial theories until a best complete theory is found. Predictions are handled with *monitors*, which are dormant processing requests passively waiting for expected constituents, and *proposals*, which are elementary hypotheses that are to be evaluated against the input. When a monitor is triggered an *event* is created calling for the evaluation of a new hypothesis. As a number of events are likely to be competing for processin at any moment a great deal of attention has been given to the problem of assigning them priorities. The HWIM system consists of several components coordinated by a control strategy using: 1) a Left-Hybrid policy to constrain the formation of seed events and their extension, 2) a Verify-at-Pick method of employing the Verification component, and 3) a Shortfall Density method of computing priority scores for ordering the event cue. On a final test involving 124 utterances (3 male speakers) 44% of the utterances were correctly understood (though the system was not fully debugged nor finely tuned at the time). Appendices include a sample set of sentence types and a sample trace of an utterance being processed.

GENERAL: ARTIFICIAL INTELLIGENCE

Research at Yale in Natural Language Processing 1976

Roger C. Schank
Department of Computer Science, Yale University, New Haven, Connecticut

Research Report No. 84, 30 pp, 1976

1. SAM - A script-based story understanding program. 2. FRUMP - A fast program designed to skim a newspaper looking for events in which it is interested. 3. PAM - A plan based program designed to understand stories that call upon general knowledge of human goals and relationships. 4. TALESPIN - A program intended to make stories to tell in an interactive mode. 5. WEIS/POLITICS - This is a program designed to read newspaper headlines and do two possible things: 1) It codes the sentences into a political conding scheme used by political scientists; 2) It simulates a person with an ideological belief system being informed of the event in the headlines. The program is then capable of answering questions based on its belief system about appropriate responses of the U.S. to the new events.

GENERAL: ARTIFICIAL INTELLIGENCE

Methodological Questions About Artificial Intelligence: Approaches to Understanding Natural Language

Yorick Wilks
University of Essex, Colchester

Journal of Pragmatics 1: 69-84, 1977

I. AI research on natural language is best regarded as an engineering activity rather than a scientific one; hence attempts to justify AI research on NL by appeal to the methods of the sciences are in general misguided. II. Semantic primitives cannot be justified in the way that *theoretical objects* in the sciences (such as neutrinos) are. Such primitives are not essentially different from the surface words whose meanings they are used to express.

Directed Recursive Labelnode Hypergraphs: A New Representation Language

Harold Boley
Institut fuer Informatik, Universitaet Hamburg, Schlueterstr. 70, 2000 Hamburg 13

Artificial Intelligence 9: 49-85, August 1977

Directed recursive labelnode hypergraphs (DRLHs) combine 3 generalizations of directed labeled graphs: 1) *Hypergraphs* which have *hyperarcs* connecting 1,2,3 . . . n nodes. 2) *Recursive graphs* which, in addition to atomic nodes, have *complex nodes*, i.e. other recursive graphs that may have contact nodes. 3) *Labelnode graphs* which, instead of distinguishing labels and nodes, have only one set of labelnodes, each of whose members may function as a label and/or as a node. DRLHs are discussed in relation to the predicate calculus, relational DBs (Codd), LISP, simple semantic networks, active structural networks (LNR group). The analysis of NL strings into DRLHs and their processing is done with pattern-matching rules.

Artificial Intelligence - A Personal View

D. Marr
Artificial Intelligence Laboratory, MIT, 545 Technology Sq., Cambridge, MA 02139

Artificial Intelligence 9: 37-48, August 1977

A theory of computation for a given problem is concerned with *what* is to be computed and *why*, while an implementation algorithm concerns *how* to do it. There are likely to be many algorithms for a given computation, with choice of one over others being highly dependent on hardware considerations. But once a computational theory has been established for a domain it need never be done again. A problem which can be decomposed into a computational theory plus algorithmic implementation is a *Type 1* theory. Some problems, especially those involving simultaneous action of a considerable number of processes, may not have a Type 1 theory. These will require a *Type 2* theory. Most AI programs have been Type 1 theory or not, and Type 2 theories may obscure the correct Type 1 decomposition of the problem. Topics discussed include Schank's CD, Newell and Simon's production systems, Norman and Rumelhart's active structural network, AI vision work. It is suggested that NL may have no Type 1 decomposition.

Computational and Mathematical Linguistics: Proceedings of the International Conference on Computational Linguistics

Florence: Leo S. Olschki Editore, 793 pp. in 2 Volumes, 1977

CONTENTS

LEXICOLOGY

TABLE OF CONTENTS VII

TEXT CORPUS EDITING

SEMANTICAL CALCULUS

VIII TABLE OF CONTENTS

*Abstracts of these papers appeared on AJCL Microfiche 1.

Statistical Methods in Linguistics, 1976

SMLNA

CONTENTS

*Abstracts of these articles have appeared previously in AJCL, Fiche 48. The specific frame numbers are indicated in parentheses after the title. The other articles are abstracted elsewhere on this fiche.

GENERAL: JOURNAL

The Prague Bulletin of Mathematical Linguistics 25

Universita Karlova, Praha, 1976

CONTENTS

GENERAL: JOURNAL

The Prague Bulletin of Mathematical Linguistics 26

Universita Karlova, Praha 1976

CONTENTS

PHONETICS-PHONOLOGY

Speech Understanding Systems: Final Report, November 1974 - October 1976 Volume II: *Acoustic Front End*

William A. Woods, *et al.*
Bolt Beranek and Newman Inc., 50 Moulton St., Cambridge, MA 02138

Report 3438, 91 pp., December 1976

The initial signal processing component computes the following types of parameters: 1) zero crossings, 2) LP analysis (in the 0-5 kHz region), 3) spectral energy, 4) formant extraction, 5) fundamental frequency. Acoustic-Phonetic Recognition (APR) consists of: SEGMENTATION, which employs a segment lattice to handle alternative segmentations, LABELING, which arrives at a rough phonetic characterization of each segment, and SCORING, to determine a score for the correspondence of each phoneme possibility for each segment. The speech synthesis-by-rule progrsm is used for response generation and, more importantly, for word verification. The phonological component makes use of syntactic and lexical information (and, potentially, semantic information) and outputs to the phonetic component. The verification component contains an analysis-by-synthesis procedure to overcome inaccuracies present in preliminary phonetic analysis and to take account of the effects of the phonological rules. Appendices: Dictionary Phonemes, APR labels, APR rules, Parameters for Scoring.

PHONETICS-PHONOLOGY: PHONOLOGY

Allophonic Variations of Stop Consonants in a Speech Synthesis-by-Rule Program

W. A. Ainsworth, and J. B. Millar
Department of Communication, University of Keele, Keele, Staffs., U.K.

International Journal of Man-Machine Studies 8: 159-168, March 1976

A computer program for synthesizing speech by rule from phonetic data has been modified so that the rules for generating stop consonants (/b, d, g, p, t, k/) depend on context. Listening tests have shown that this expedient can increase the intelligibility of stop consonants in isolated CV syllables from 68% to 92%, with about 3 allophones per consonant being required to achieve this level of performance.

WHY "/ɪ/book"?

Bengt Orestrom
Survey of Spoken English, University of Lund, Helgonabacken 14, S-223 62 Lund, Sweden

SSE, 13 pp., October 1976

It seems easiest to explain the use of /ɪ/ before a consonant as a hesitation phenomenon. Unlike the regular, hesitational long /ɪ:/, the short /ɪ/ does not appear to be immediately due to lexical selection (groping for the right word) but is rather sparked off occasionally in a hesitational surroundng. However, a broader study is needed of the whole discourse situation to specify more exactly what constitutes a hesitational surrounding. Moreover, there are probably other factors than lexical selection and planning that might cause hesitation, such a frequent interruptions by another speaker.

WRITING: RECOGNITION: HANDWRITTEN CHARACTERS

Using Knowledge in the Computer Interpretation of Handwritten FORTRAN coding Sheets

R. Bornat, and J. M. Brody
Computing Centre, University of Essex, Colchester, U. K.

International Journal of Man-Machine Studies 8: 13-27, January 1976

It is argued that, in order to construct a computer program which can successfully read casually hand-printed FORTRAN coding sheets, it is necessary to use processes which explicitly incorporate consistency checks based on our knowledge of how handprinted characters must be constructed, how FORTRAN statements must be written and of how statements must be put together to form a program. That is, we propose to treat FORTRAN coding sheets as the object "world" of an AI program. This paper gives a brief description and justification of the proposed methodology, including an argument that this approach will not lead to a combinatorial explosion of search time.

A Scrabble Crossword Game Playing Program

Stuart C. Shapiro
Department of Computer Science, SUNY at Buffalo, Amherst, NY 14226

Howard R. Smith
Department of Computer Sciences, The University of Texas at Austin, 78712

Department of Computer Science, SUNY at Buffalo, Technical Report 119, March 21, 1977

A program has been designed and implemented in SIMULA 67 on a DECSystem-10 to play the SCRABBLE Crossword Game interactively against a human opponent. The heart of the design is the data structure for the lexicon and the algorithm for searching it. The lexicon is represented as a letter table, or tree using canonical ordering of the letters in the words rather than the original spelling. The algorithm takes the tree and a collection of letters, including blanks, and in a single backtrack search of the tree finds all words that can be formed from any combination and permutations of the letters. Words using the higher valued letters are found before words not using those letters, and words using a collection of letters are found before words using a sub-collection of them. The Search procedure detaches after each group of words is found and may be resumed if more words are desired.

Speech Understanding Systems: Final Report, November 1974 - October 1976 Volume III: *Lexicon, Lexical Retrieval and Control*

William A. Woods, *et al.*
Bolt Beranek and Newman Inc., 50 Moulton Street, Cambridge, MA 02138

Report No. 3438, 110 pp., December 1976

Topics: Dictionary, Phonological rules, Dictionary expansions, Lexical retrieval, Control strategy, Performance. The Lexical Retrieval component determines the n most probable word matches in a full lexicon or appropriate subset and operates on a phonetic segment lattice. Words can be matched left-to-right and right-to-left. Control strategy options are governed by 25 flags. Each strategy performs an initial scan of some region of the utterance, creating one-word seed events. In "middle-out" strategies the initial scan is done over the entire utterance. In L-R strategies the initial scan only considers words that could begin the utterance. In "hybrid" strategies the initial scan fixes on an initial portion of the utterance and then middle-out analysis is done on this region with the remainder necessarily being analyzed L-R. In all these options events are ordered on the queue by their priority scores. Appendices: Annotated phonological rules, Format and examples of dictionary files, Result summaries for each token, Performance results for strategy variations, BIGDICT and TRAVELDICT listings, Dictionary expansion - a user's guide.

LEXICOGRAPHY-LEXICOLOGY: DICTIONARY

Computers and the Production of Systematic Terminological Glossaries

M. L. Hann
UMIST, Manchester, England

Bulletin of the Association for Literary and Linguistic Computing 5: 26-37, 1977

It would be useful if specialized terminological glossaries were organized in a way which reveals the conceptual structure of the domain glossed rather than being organized alphabetically. For the purpose of formal description a terminological relation can be considered as consisting of two component relations: a logical and an ontological relation. Logical relations are abstract relations and as such are expressible in terms of the usual set-theoretic predicates, whereas ontological relations are concrete and must be defined in terms of real world criteria. The basic notion is elaborated and then ilustrated through application to the domain of computing terminology.

LEXICOGRAPHY-LEXICOLOGY: TEXT HANDLING

SITAR: Interactive Text Processing System for Small Computers

Ben Ross Schneider, Jr.
English Department, Lawrence University, Appleton, WI 54911

Reid M. Watts
University of Kansas

Communications of the ACM 20: 495-499, July 1977

SITAR, a low-cost interactive text handling and text analysis system for nontechnical users, is in many ways comparable to interactive bibliographic search and retrieval systems, but has several additions features. It is implemented on a PDP/11 time-sharing computer invoked by a CRT with microprogrammed editing functions. It uses a simple command language designating a function, a file, and a search template consisting of the textual string desired and strings delimiting the context in which the hit is to be delivered.

Manual for Terminal Input of Spoken English Material

Bengt Orestrom, Jan Svartvik, and Cecilia Thavenius
Survey of Spoken English, University of Lund, Helgonabacken 14, S-223 62 Lund, Sweden

SSE, 23 pp., October 1976

The basic material of the Survey of Spoken English (SSE) at Lund University consists of the spoken English at the Survey of English material collected and transcribed at the Survey of English Usage (SEU) under the direction of Professor Randolph Quirk, University College London. The SSE material differs from the original SEU material in two respects: for linguistic reasons (explained in the manual) it has been pruned of certain prosodic and paralinguistic categories; and for technical reasons the prosodic and phonetic symbols have been changed. The manual lists the symbol set used for SSE and code set, peripheral equipment, and SSE procedures have been described in appendices.

Chapter III: The Executive System

William H. Paxton
Speech Research Institute, Menlo Park, CA 94025

Speech Understanding Research Final Technical Report, 15 October 1975 - 14 October 1976, III-1 to III-108, October 1976

The Executive in the speech understanding system has three main tasks: 1) it coordinates the work of the other components of the system by calling acoustic processes and applying language definition procedures, 2) it assigns priorities to the various tasks in the system, and 3) it organizes results so that information is shared and duplication of effort is avoided. The Executive performs a series of tasks to find words in the speech signal and to organize them into phrases of the input language with the ultimate goal of creating a root category phrase that spans the input. Thus, because the system has been designed with the language definition as the primary mechanism for specifying knowledge source interactions, the Executive does the job of a parser in fulfilling its responsibilities for system integration and control. The main Executive data structure is the 'parse net' whose nodes are either 'phrases' or 'predictions.' Two main types of tasks: *predict*, which operates top-down, *word*, bottom-up.

GRAMMAR: PARSER

Analyzing English Noun Groups for their Conceptual Content

Anatole V. Gershman
Department of Computer Science, Yale University, New Haven, CT 06520

Research Report No. 110, 37 pp., May 1977

The Noun Group Processor (NGP) is a set of programs which is an integral part of ELI, the English Language Interpreter (Reisbeck and Schank 1976, AJCL 64:57). It is a production-like system which uses expectation as its basic control mechanism. Input is processed one word at a time, from left to right, using linguistic and world knowledge to parse noun groups into the Conceptual Dependency representation. Four classes of noun groups are differentiated on the basis of the semantic structures they generate: 1) Picture Producers, 2) Concept Producers, 3) Time Descriptors, 4) State Descriptors. Dictionary entries for individual words contain much of the program's knowledge. In addition, a limited ability for processing slightly incorrect sentences and unknown words is incorporated.

Finite-State Parsing of Phrase-Structure Languages and the Status of Readjustment Rules in Grammar

D. Terence Langendoen
Ph. D. Program in Linguistics, C.U.N.Y. Graduate Center, 33 West 42nd St., New York, 10036

Linguistic Inquiry 6: 533-554, Fall 1975

A Minimally Augmented Finite Parser (MAFP) is proposed as a model of linguistic performance and an algorithm for the construction of a MAFP for any normal-form grammar is presented. Since MAFP's are minimally augmented their use of the extra power of a PDS (push-down store) is limited to just those situations in which that power is necessary for effective parsing. They do not need to use their PDS to parse all noun phrases within sentences. The readjustment of the surface structure of sentences with LE or RE greater than 3 or 4 suggests that the RE and LE structures are unacceptable and that unacceptability is due to the limited amount of PDS available to the MAFP in the human sentence recognition device. It is argued that Readjustment Rules (whose formal structure is quite different from that of transformations) belong to a distinct component of the grammar which relates syntax and phonology and a readjustment rule schema is suggested which must be universal.

Speech Understandng Systems: Final Report, November 1974 -October. Volume IV *Syntax and Semantics*

William A. Woods, *et al.*
Bolt Beranek and Newman, Inc., 50 Moulton St., Cambridge, MA 02138

Report No. 3438, 132 pp., December 1976

HWIM uses a middle-out, bi-directional ATN parsing algorithm. There are four current HWIM grammars - BIGGRAM, MIDGRAM, SMALLGRAM, SPEECHGRAMMAR - and they use five arc types: PUSH, POP, WRD (word), CAT (category), and JUMP. The first grammar (SPEECHGRAMMAR) processed words by their usual parts of speech and constructed ordinary syntactic parse trees for a wide variety of complex constructions. The newer grammars use "semantic" categories on their arcs as well as the traditional syntactic ones; and BIGGRAM and MIDGRAM also incorporate pragmatic information. Prosodic information is handled by marking those arcs expected to be accompanied by prosodic boundaries. Appendices: Listing of MIDGRAM Grammar, Sample Parse-Interpretations, Parser Trace.

Chapter XIII: Generating Verbal Responses

Jonathan Slocum
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975-14 October 1976, XIII-1 to XIII-11, October 1976

Verb templates and noun templates are associated with specific word senses (prototypical nodes) in the net and serve to order the constituents in a phrase and to indicate how each constituent is to be generated by naming a function to be called with the network constituent. For generation purposes at least this modular grammar has an important advantage over a 'monolithic' grammar: it clearly indicates the syntactic idiosyncracies imposed by particular word choices. The storage requirements of the two formalisms are probably similar. The modular grammar will probably require more rules, but a monolithic grammar must in turn incorporate many applicability tests for each of its rules. In effect, these tests are precomputed during the construction of the modular grammar.

GRAMMAR: GENERATOR

Towards a Model of Language Production: Linguistic and Computational Foundations

Henry Thompson
Department of Linguistics, University of California at Berkeley, Xerox Palo Alto Research Center

Statistical Methods in Linguistics 1976: 110-126

Computational approaches to language analysis are quite sophisticated while computational understanding of language generation is all but non-existent. To get out of this impasse attention must be given to matters of rhetoric and lexical semantics. The work of Halliday (given/new, theme/rheme), Sgall, and Fillmore (schemata, frame) is of particular relevance in dealing with these issues. The essay concludes with a discussion of how these problems might be modeled in the Knowledge Representation Language (KRL) being developed by Bobrow and Winograd.

GRAMMAR: CLASSES & CONSTRUCTIONS

Verbs, Time, and Modality

M. J. Steedman
University of Sussex

Cognitive Science 1: 216-234, July 1977

Vendler's classification of verbs into aspectual categories (activities, accomplishments, achievements, and states) is better seen as classifying the meanings of sentences, and a recursive scheme is then applied to the modal verbs *must*, *will*, and *may*. Of *will* and *may* it is concluded: 1) that these models have two and only two senses - an epistemic and a deontic one; 2) that the epistemic sense is found with "situations" as its complement propositions, while the deontic is found only with "events"; 3) the future tense of certain sentences involving epistemic modals arises from the status of the complement as a "prospective situation" (as defined in Section I), paraphraseable using "be going to." These postulates alone are both necessary and sufficient to explain *may*.

GRAMMAR: CLASSES & CONSTRUCTIONS

The Interpretation of Temporal Order in Coordinate Conjunction

Martin S. Chodorow, and Lance A. Miller
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6199, 21 pp., September 13, 1976

This paper provides a non-contextual analysis of the temporal order of actions that are expressed as coordinately conjoined verbs. Although the analysis moves in the domain of cooking recipes, its principles seem generally applicable. Cooking instructions containing a pair of conjoined verbs are interpreted as requiring the two actions to be performed either consecutively or simultaneously. If the actions are compatible, they *may* be executed simultaneously; if they are incompatible, they *must* be executed consecutively. Compatibility is defined in terms of preconditions and on-going conditions for actions. Consecutive actions are often accompanied by interaction effects which can be attributed to partially or incorrectly fulfilled preconditions. Preconditions and compatibility provide the framework for a sufficient solution to one type of interaction problem. The set of pre- and on-going conditions for an action is entailed by the verb which expresses that action. This entailment relationship is consistent with the general requirements for a non-contextual solution to the problem.

On Keenan's Definition of 'Subject of'

David E. Johnson
Mathematical Sciences Department, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6035, 46 pp., June 7, 1976

E. L. Keenan has attempted to provide a universal definition of the natural language notion 'subject of' in terms of the syntactic, semantic and progmatic properties typically assoicated with noun phrases functioning as subjects. Keenan's proposal is inadequate as a formal definition in a number of crucial aspects, the *Subjects Properties List* (SPL) is poorly motivated, the notion of 'clear preponderance' of SPL properties is not clear. Keenan's results are more reasonably interpreted as providing a *reductio ad absurdum* argument in favor of the position that grammatical relations such as 'subject of' be adopted as primitive theoretical terms in linguistic theory rather than being defined terms. Keenan's approach might, however, prove extremely useful as a heuristic method for isolating potential subjects, e.g. field work situations.

SEMANTICS-DISCOURSE

Speech Understanding Systems: Final Report, November 1974 - October 1976. Volume V: Trip

William A. Woods, *et al.*
Bolt Beranek and Newman, Inc., 50 Moulton St., Cambridge, MA 02136

Report No. 3438, 115 pp., December 1976

The travel budget manager's assistant facilitates NL communication between man and machine in a number of ways, including: 1) The state of the discourse and the data base are made available to the speech understander while processing an utterance to constrain its possible interpretations. 2) Inference mechanisms make possible looser expression of questions and commands by the manager by divorcing semantic interpretations from explicit data structures. 3) English-like responses are used both to give explicitly requested information and to exhibit the program's state of knowledge regarding the on-going dialogue. Many types of knowledge are required for this task, including linguistic knowledge, discourse knowledge, and various types of world knowledge. Knowledge which is relatively fixed (such as syntactic knowledge) is stored in the ATN, while relatively fluid knowledge (e.g. data about a specific trip) is represented in the semantic network. Tests on ATN arcs link these two representations.

SEMANTICS-DISCOURSE: THEORY

Chapter VI: The Model of the Domain

Gary G. Hendrix
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975 - 14 October 1976, VI-1 to VI-13, October 1976.

Using a partitioned semantic network, information about ships in the U.S., Soviet and British fleets was incorporated into a database: 76 classes of ships, 740 individual ships, over 200 known by name. Characteristics encoded include: owner, builder, length, beam, draft, displacement, number in crew, speeds (surface and submerged), class, type. At the top level this domain model is encoded as a large conjunction of individual facts and general rules. The UNIVERSAL set is divided into seven major disjoint subsets: UNITS OF MEASURE, MEASURES, NUMBERS, PHY.LP (whose principle purpose is to show the relationships between physical objects, inanimate physical objects, and legal persons), PROCEDURES, SITUTATIONS, SHIP GROUPS.

SEMANTICS-DISCOURSE: THEORY

Chapter V: The Representation of Semantic Knowledge

Gary G. Hendrix
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975 - 14 October 1976, V-1 to V-99, October 1976

A partitioned network consists of nodes and arcs which are partitioned into spaces which group information into bundles. Spaces can be combined into vistas. Most operations are performed from the vantage of one of these vistas with the effect that the operations behave as if the entire network were composed solely of those nodes and arcs that lie in the spaces of a given vista; all else is ignored. When necessary, spaces can be given all the properties normally associated with nodes. In particular, arcs from ordinary nodes may point to such spaces (which are called supernodes). Supernodes are primarily used for encoding higher-order structures, including logical connectives, quantification, and questions.

SEMANTICS-DISCOURSE: THEORY

What Sort of Taxonomy of Causation Do We Need for Language Understanding?

Yorick Wilks
University of Edinburgh

Cognitive Science 1: 235 - 264, July 1977

A taxonomy of causes beyond a distinction between CAUSE and GOAL is unnecessary. And even this distinction is largely functional, in many cases being reduced to no more than the directionality of a rule. In the case of human and human-like actions we should first explore the GOAL "explanation" of an action whereas the CAUSE "explanation" should be explored first with nonhuman events. There seems to be no reason to think that a more complex taxonomy of causes (beyond causes and reasons (GOAL)) is required, at least not if, as here, one seeks a procedural distinction corresponding to any taxonomic distinction between inference rules. Finally, the preference semantics treatment is compared with positions established by Charniak, Rieger, and Schank.

SEMANTICS-DISCOURSE: THEORY

Dialogue Games: Meta-communication Structures for Natural Language Interaction

James A. Levin, and James A. Moore
USC Information Sciences Instutute, 4647 Admiralty Way, Marina del Rey, Ca 90291

Report No. ISI/RR-77-53, 36 pp., January 1977

People in dialogue interact according to established patterns which span several turns in a dialogue and which recur frequently. These patterns appear to be organized around the goals which the dialogue serves for each participant. *Dialogue-games* are structures which model these patterns. A Dialogue-game has *Parameters*, which represent those elements that vary across instances of a particular pattern - the particular dialogue participants and the content logic. The states of the world which must be in effect for a particular Dialogue-game to be employed successfully are represented by *Specifications* of these Parameters. Finally, the expected sequence of intermediate states that occur during instances of a particular conventional pattern are represented by the *Components* of the Dialogue-game. Reresentations for several Dialogue-games are presented and a process model is discussed.

A Goal-Oriented Model of Natural Language Interaction

James A. Moore, William C. Mann, and James A. Levin
USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90291

Report No. ISI/RR-77-52, 61., January 1977

This report gives an overview of the ISI research program in modeling human communication. The methodology involves selecting a single, naturally-occurring dialogue, instructing a human observer to extract certain aspects of the dialogue relating to its comprehension, and then usng these aspects to guide the building and verification of a model of the dialogue participants. According to the model people use language to pursue their own goals and are able to communicate effectively because they share an understanding of a collection of interrelated, cooperative goal structures. This report contains a detailed statement of the problem, a review of related research, and a description of the contributions of this research to linguistic theory. The current state of the dialogue model is described (with a detailed simulation in an Appendix) and the general research methodology is explored. Finally, the deficiencies of existing man-machine interfaces are summarized.

On the Referential Attributive Distinction

Eugene Charniak
Institut pour les Etudes Semantiques et Cognitives, Universite de Geneve

Working Paper 24, 55 pp., 1976

According to Donnelan the phrase *Smith's murderer* in the sentence *Smith's murderer is insane* is being used referentially when the sentence is being used to describe someone (e.g. Carmen Jones) known to have murdered Smith while it is being used attributively if the identity of the murderer is unknown but there is reason to believe that person to be insane. Close analysis reveals that the referential/attributive distinction is not primitive, but rather can be derived from more basic considerations. The rules which account for the phenomena which distinguish referential and attributive uses do not themselves depend on the distinction. Hence there is no reason to include the distinction in semantic representation, or any other level of representation for that matter.

What's in a Concept: Structural Foundations for Semantic Networks

Ronald J. Brachman
Harvard University, Cambridge, MA 02138 Bolt Beranek and Newman, Inc. 50 Moulton St., Cambridge, MA 02138

International Journal of Man-Machine Studies 9: 127-152, 1977

That semantic nets have failed to live up to their originally perceived promise is due to inadequate consideration of their foundations. Many of the links used in semantic networks have hidden import - while concept nodes look like they represent classes of objects, they are most often expected to represent much more than that. To remedy the situation network notations require a level of epistemological representation that explicitly accounts for operations such as attribute-description, structure-passing, and attribute/value-binding. Such a level has been embodied in a set of primitive link types, on top of which concepts can be built in a well-specified, consistent matter through the use of devices for specifying the structural interrelations between attributes. The "constellation" of links formed by the attribute definitions and structuring Gestalt is really a *structure* for the node. Procedures for using the foundation to automatically build instances and conceptual modifications are presented and the intensional nature of such a representation and its implications are discussed.

The Need for a Frame Semantics within Linguistics

Charles J. Fillmore
Department of Linguistics, University of California, Berkeley, CA 94720

Statistical Methods in Linguistics 1976: 5-29

We should think of the representation of the meaning of a word or text as a set of instructions addressed to a cartoonist or a film-maker, these instructions imposing constraints on how a comic strip or film strip or movie can be made which will display an image or situation representing what the word or text can "mean." These representations will have to deal with: time schemata (simultaneity, sequence, span, calendar), perspectives (point of view), sequences of inferences, cross-referencing and embedding (scenes within scenes in relations of thinking, knowing, wanting, etc.), background information, provisions for indexicality. Many examples are given and the view is advanced that much language is of a highly formulaic nature (involving little or no 'creativity' or 'free generation') and keyed to specific contexts.

Meaning of Sign, Cognitive Content, and Pragmatics

Petr Sgall
Centre for Numerical Mathematics, Charles University, Prague, Czechoslovakia

The Prague Bulletin of Mathematical Linguistics 25: 51-68, 1976

Natural language is studied with widely different theoretical frameworks on the basis of the binary structure of the sign or of the corresponding operation, a general mapping, with the so-called "asymmetrical dualism," into several steps which yield the well-known patterns of levels linearly ordered from semantics to phonetics. But this articulation is only the first stage of a necessary structuring of the vast domain of semantics. There are no finished solutions as yet, but it seems that the distinction of meaning and content does not belong to the quoted ordering of levels, since they all belong to the linguistic system while the cognitive content does not. Nevertheless, assuming the existence of a "universal" logical language - including the advantages and richness of higher order predicate calculus, etc. - the relationship between meaning (or language as such) and cognition might be accounted for in the form of translation procedures (which do not necessarily presuppose unambiguous content).

SEMANTICS-DISCOURSE: THEORY

Types of Processes on Cognitive Networks

David G. Hays
Department of Linguistics, SUNY at Buffalo, Amherst, NY 14260

Computational and Mathematical Linguistics: Proceedings of the International Conference on Computational Linguistics, Leo S. Olschki Editore: 523-532, 1977

The model has four types of nodes: Events, Entities, Properties, Modalities (which is a spatiotemporal localization). There are five major types of arcs, with some subtypes: Paradigmatic, Syntagmatic, Discursive, Attitudinal, and Metalingual. Discursive arcs link modalities together while attitudinal arcs link discourse to entities. Metalingual arcs links a node to a modality so that the network pattern under the modality becomes a definition of the metalingually related node; this process is recursive. PATH TRACING along arcs is well understood and fairly simple, perhaps requiring no more than a finite-state automaton for the job. The paths traced can be homogeneous, involving only one arc type (such as general zation or specialization on paradigmatic arcs), or heterogeneous, involving several arc types. However, since language works on and cognitive networks are related to language in such a way that they must work on themselves, we must also be able to do PATTERN MATCHING in cognitive networks, which is considerably more difficult than path tracing. A fragment of a network can be a template or program that operates on another part of the same network so that, for example, a pattern under a new modality can be matched to a pattern already in the network and thereby establish the new pattern as an instance of the old.

SEMANTICS-DISCOURSE: THEORY

Several Ways to be Suggestive: An Examination of Presupposition

Margaret King
Institut pour les Etudes Semantiques et Cognitives, Universite de Geneve

Working Paper 25, 63 pp., 1977

An examination of the literature on presupposition leads to the rejection of the notion of presuppostition as a unitary phenomenon. Nonetheless it seems clear that some sentences do, with varying degrees of strength, make suggestions. Three main classes of suggestion mechanisms are considered: 1) Suggestions depending on phrase forms, 2) Suggestions depending on a sentence-form, 3) Suggestions carried by particular lexical items. The use of such mechanisms of NL understandng systems is considered.

SEMANTICS-DISCOURSE: COMPREHENSION

Mechanizing Temporal Knowledge

Kenneth Kan, and G. Anthony Gorry
MIT, Cambridge, MA 02139

Artificial Intelligence 9: 87-108, August 1977

A time specialist is a program which is knowledgeable about time in general and which can be used by a higher level program. The time specialist can deal with different kinds of temporal specifications, incuding fuzzy ones (e.g. "a few weeks ago"): 1) events organized by *dates*, 2) in terms of *special reference events* (e.g. "birth," "now"), 3) *before/after chains*. Three basic types of questions can be answered: 1) Did X happen at T? 2) When did X happen? 3) What happened at T? Database consistency and error correction are discussed and the speciaist's treatment of a time-travel story by Robert Heinlein is presented.

SEMANTICS-DISCOURSE: COMPREHENSION: SYSTEM

Chapter X: Ellipsis

Barbara J. Grosz
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975-14 October 1976, X-1 to X-32, October 1976

Ellipsis can occur at the sentence or clausal level, or at the level of NP or VP; this chapter concentrates on the sentence level. Building an interpretation of an elliptical phrase entails two steps once ellipsis has been detected. 1. *Slot Determination.* Items missing from the utterance must be found in the preceding utterance (the slot the phrase fills in the preceding utterance must be determined). 2. *Expanding the Utterance.* A complete phrase must be built using the elliptical phrase and the missing constituents found in the previous utterance.

SEMANTICS-DISCOURSE: COMPREHENSION: SYSTEM

Chapter IX: Resolving Definite Noun Phrases

Barbara J. Grosz
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975-14 October 1976, IX-1 to IX-32, October 1976

Comprehension entails identifying old concepts in memory and attaching new information to them. Definite noun phrases (DEFNPs) are the most frequently used means of expressing old information. Context plays a crucial role in identifying the reference of DEFNPs and the system uses a focus space partition to represent this context. While the resolution of both pronouns and nonpronominal DEFNPs use global dialog context and immediate context, the former is more important for DEFNPs, the latter for pronouns. The resolution procedures all depend on the existence of a representation of focus of attention.

SEMANTICS-DISCOURSE: COMPREHENSION: SYSTEM

Chapter VIII: Discourse Analysis

Barbara J. Grosz
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975-14 October 1976, VIII-1 to VIII-62, October 1976

The bulk of the chapter concerns the collection and analysis of two types of dialogs: 1) task-oriented dialogs involving communication between two people cooperating to complete a task; 2) data-base-oriented dialogs involving communication directed toward obtaining information from a computer data base. Diaog analysis reveals that contextual influences operate at two levels in a discourse. 1) The global context - the total discourse and situational setting - provides a set of constraints which the system uses for the resolution of definite noun phrases by partitioning the network into focus spaces. 2) The immediate context of closely preceding utterances is used in the interpretation of elliptical expressions.

SEMANTICS-DISCOURSE: COMPREHENSION: SYSTEM

Chapter VII: Semantic Aspects of Translation

Gary G. Hendrix
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975-14 October 1976, VII-1 to VII-59, October 1976

The semantic component can perform three functions: 1) It may filter out phrase combinations which do not meet semantic criteria. 2) For combinations that are acceptable the semantic component may build deep, internal structure representing the meaning of the input (or portions of it) in the context of a particular task domain. As filtering (by both semantics and discourse) is dependent upon the structures assigned to subphrases of the input, filtering and structure building are combined. If any of various checks and restrictions in the structure-building process recognize an anomalous condition in a structure being built, then the structure building fails, and this failure, acting as a filter, serves to reject the phrase combination. 3) The semantic component may make predictions concerning what words or syntactic contructions are likely to occur in other parts of the utterance. This chapter emphasizes 1 and 2, with only a brief treatment of 3.

SEMANTICS-DISCOURSE: COMPREHENSION: SYSTEM

GUS, a Frame-driven Dialog System

Daniel G. Bobrow, Ronald M. Kaplan, Martin Kay, Donald A. Norman, Henry Thompson, and Terry Winograd
Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304

Artificial Intelligence 8: 155-173, April 1977

GUS is a dialogue system designed to act as a travel agent. It can tolerate mixed initiative dialogue and its expectations about future input evolve in the course of conversation. The system consists of a morphological analyzer, a syntactic analyzer, frame reasoner, and language generator, all tied together by an overall asynchronous control mechanism. The notion of frame implemented in the system bears a family resemblance to Minsky's notion, but the relationship is only that, familial. A *frame* consists of a *name* (which is primarily a mnemonic device for the system builders), a reference to a *prototype* frame, and a set of *slots*. A prototype frame is a template for its *instances*. Slots consist of a *slot-name*, a *filler* or *value* and possibly a set of attached procedures. The value of a slot may be another frame. To conduct a dialogue the system creates an instance of a dialogue frame and begins to fill slots for the instance in accordance with the specifications in the prototype.

SEMANTICS-DISCOURSE COMPREHENSION: CONCEPTUAL DEPENDENCY

Skimming Newspaper Stories by Computer

Gerald F. DeJong
Department of Computer Science, Yale University, New Haven, CT 06520

Research Report No. 104, 31 pp., May 1977

FRUMP (Fast Reading Understanding and Memory Program) is designed to understand newspaper stories and was designed to overcome some of the problems that arose from the SAM system over the last several years. Like SAM, FRUMP is a script based understander. But FRUMP is a newspaper skimming program, using 'sketchy' scripts rather than full scripts, rather than a program that carefully reads text. In reading stories FRUMP decides whether the stories are new or updates of news events that it has already seen, and stores the important information from the article. FRUMP can then give information on a news event by means of different length summaries. FRUMP can understand and produce a brief summary of a 150 word news article taken directly from a newspaper in about 5 seconds of CPU time on a DEC KA10 processor.

How to Connect Frames

Manfred Wettler
Institut pour les Etudes Semantiques et Cognitives, Universite de Geneve

Working Paper 29, 54 pp., 1976

The problem with frame-like systems is that, on the one hand we have to find some mechanisms which give the system some flexibility if prerequisites nor subacts of frames are violated. On the other hand, neither a taxonomy of these prerequisites or substates, as proposed by Charniak, nor the addition of "What-If"-rules form satisfying solutions. A possible solution to this problem consists in the formulation of general rules describing why certain prerequisites or substates have to be achieved. These rules should be attached to frames or frame-statements but they should be used only in cases where the frame alone does not allow sufficient understanding. Toward this end five new links between frames are introduced: normally produced for, participants in, reason, caused and caused by, while. Ten inference rules of three types are proposed. The first type of rule helps to decide why the actions mentioned in the text have been accomplished. The second type of rule is used to figure out how those actions or substates might have been achieved. The third type allows prediction of consequences of actions and states.

SEMANTICS-DISCOURSE: MEMORY

Natural Language Understanding Based on a Freely Associated Learned Memory Net

Sara R. Jordan
Computer Science Department, University of Tennessee, Knoxville

International Journal of Computer and Information Sciences 6: 9-25, March 1977

METQA (MEchanical Translation and Question Answering) accepts unsegmented input strings of NL from a human trainer and, after processing each string, outputs a NL response. The built-in structure of METQA consists of: 1) the capacity to build a network of nodes and labeled arcs, and 2) the general procedure of categorizing memory nodes into classes according to their behavior and usage. Link types: transform, combination, description, class, membership/subset, equivalence. Each node is a list representing a state of some word or phrase during its processing. The structure of the memory itself is independent of the subject matter. METQA learns by comparing response output with any (specially marked) feedback string the trainer may give. The program then determines which, if any, portion of the original input string was processed incorrectly and appropriate memory modifications are made.

SEMANTICS-DISCOURSE: MEMORY

Ms. Malaprop, A Language Comprehension Program

Eugene Charniak
Institut pour les Études Semantiques et Cognitives, Université de Geneve

Working Paper 31, 22 pp., 1976

Ms. Malaprop is being designed to answer questions about simple stories dealing with painting, with stories, questions, and answers being expressed in semantic representation, rather than in English. The commonsense knowledge needed to accomplish the task is provided by a frame representation (described in ISCS 29). After reviewing this representation the following issues are discussed: depth of the representation, search and pattern matching, guessing and guessing wrong.

SEMANTICS-DISCOURSE: MEMORY: CONCEPTUAL DEPENDENCY

Human and Computational Question Answering

Wendy Lehnert
Department of Computer Science, Yale University, New Haven, CT 06520

Cognitive Science 1: 47-73, January 1977

Working within the environment of Schank's SAM system the question-answering process has two parts: The *interpretive* phase takes a question in CD form and categorizes it as a *why, how, yes or no, occurrence,* or a *component* question. In the *response* phase the memory is searched for an answer in a manner appropriate to the question type. A *static* response uses information from the memory representation generated at the time the text was read while a *dynamic* response occurs when the answer must be actively reasoned by applying general world knowledge to the memory representation of the text. The overall process is also influenced by an intentionality factor which allows for variations in the mode of question answering such as detailed or sparse answers.

The Process of Question Answering

Wendy Lehnert
Department of Computer Science, Yale University, New Haven, CT 06520

Research Report No. 88, 282 pp., May 1977

A theory of NL question answering has been implemented in a computer program, QUALM, in which processing is divided into four phases: 1) Conceptual Categorization, 2) Inferential Analysis, 3) Content Specification, and 4) Retrieval Heuristics. Conceptual Categorization guides subsequent processing by dictating which specific inference mechanisms and memory retrieval strategies would be invoked in the course of answering a question. Inferential Analysis is responsible for understanding what the questioner really meant when a question should not be taken literally. Content Specification determines how much of an answer should be returned in terms of detail and elaborations. Retrievel Heuristics do the actual digging in order to extract and answer from memory. All of the inference processes within these four phases are independent of language, operating within conceptual representations.

SEMANTICS-DISCOURSE: EXPRESSION

Chapter XI: Responding on the Basis of the Semantic Translation

Gary G. Hendrix
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975-14 October 1976, XI-1 to XI-11, October 1976

The resources that may be marshalled by the responder include a component that performs logical deduction, a natural language generator, and a routine for drawing partitioned network structures. The task of the responder is to determine which inputs are requests for information that may be acted upon by the deduction component and which are not. For those that are not a representation of the corresponding partitioned network structure is drawn to express the system's interpretation of the utterance. Otherwise the responder formats a call to deduction ...d interprets the results returned by it. Depending upon the type of information requested and the results returned the responder will either produce a specified response, like YES or NO, or will invoke the English generator to express the results of the deduction processing.

A Select Bibliography of Studies in Spoken English

Cecilia Thavenius
Survey of Spoken English, University of Lund, Helgonabacken 14, S-223 62 Lund, Sweden

SSE, 14 pp., October 1976

The guiding principle when choosing items for the bibliography has been to include works which discuss the features of spoken English and how they differ from those of written English or which relate the prosodic aspect to those of grammar or communication. The bibliography is in two sections. The first section is an annotated bibliography of 10 works considered especially important; the second section lists 65 items (including the 10 from the first section).

Categorial Grammar Calculus

Hans Karlgren
Research Group for Quantitative Linguistics, Stockholm

Statistical Methods in Linguistics 1974: 1-126

This exposition of categorial grammar is intended for both mathematicians and linguists and is written accordingly. Professionals in either category should, without real effort, be able to supplement this exposition according to their professional interests: explicit proofs for a hierarchy of general theorems or explicit references to the linguistic observations motivating the proposed calculus. Main topics: 1. Formal Preliminaries, 2. Categorial Grammar Proper, 3. Generalized Categorial Grammar, 4. Multi-Index Calculus, 5. Evaluation of Categorial Expressions, 6. Fragment Analysis.

LINGUISTICS: METHODS: MATHEMATICAL

Prague Studies in Mathematical Linguistics 5

Academia, Publishing House of the Czechoslovak Academy of Sciences, 1976

CONTENTS

LINGUISTICS: METHODS: MATHEMATICAL

Some Recent Contributions to Statistical Linguistics

Charles Muller
12 Ave de la Forte Noire, 6700 Strasbourg, France

Statistical Methods in Linguistics 1976: 136-147

I: Word frequencies in Giraudoux: characteristic vocabulary of 'le roman fantastique'; themes and lexical items; vocabulary of French tragedy, lexical specialization of text segments; statistical determination of authorship. II: For a corpus of French texts totally 20,000 phonemes it has been found that consonants have a small and stable variance whereas the variance of the vowels is great with large variation. III: Indices of vocabulary richness for comparison of texts of different lengths, a procedure for estimating the number of words in a particular vocabulary which *have not* been used in a particlar text based on that vocabulary.

LINGUISTICS: METHODS: LOGIC

On Dealing with Quantification in Natural Language Utterances

Peter B. Sheridan
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6422, 44 pp., March 4, 1977

Quantifiers are treated as higher predicates. Wffs of sorted first-order logic (with descriptions) are represented in attribute-value record structures. Short of eventually taking context fully into account the surface order of quantificational terms is taken as the intended order. The fact that quantificational concepts in NL often combine a "quantificational particle" - such as 'some', 'each' 'any', etc. - with a "range term" - e.g. 'one', 'thing', 'time', 'document', etc. - is utilised in the reduction of quantificational utterances to canonical form. Also discussed: quantification and connectives in the MBA (Mentor for Business Applications) parser; ambiguities connected with 'a', 'any', and 'some'; multiple predication; anonymous quantification.

LINGUISTICS: METHODS: LOGIC

Special Issue on Fuzzy Logic I

International Journal of Man-Machine Studies 8, No. 3, 1976

CONTENTS

LINGUISTICS: METHODS: LOGIC

Special Issue on Fuzzy Logic II

International Journal of Man-Machine Studies 8, No. 6, 1976

CONTENTS

LINGUISTICS: METHODS: LOGIC

The Proper Treatment of Montague Grammars in Natural Logic and Linguistics

Asa Kasher
Department of Philosophy, Tel Aviv University, Israel

Theoretical Linguistics 2, 1/2: 133 - 145, 1975

Any wholesale comparison of generative theories which Montague grammars is, in at least one sense, pointless, since they are solutions to distinctly different problems, under extremely different conditions of adequacy. The same conclusion holds, indeed, for a full-scale comparison of natural logic theories and Montague grammars. A redirecting of the study of Montague grammars is suggested and a methodological heuristic of formal extensions paired with empirical restrictions is defended and illustrated.

LINGUISTICS: METHODS: LOGIC

Why I am not a Montague Grammarian

Richard M. Martin
Department of Philosophy, North-western University, Evanton, IL, 60201

Theoretical Linguistics 2, 1/2: 147 - 157, 1975

Much of the significance of Montague's work rests upon the acceptability of a second-order functional calculus with a modal operator, and the extensive model theory based on it. An alternative is suggested with consists of an applied, first-order logic with the calculus of individuals and event logic, and with first-order inscriptional semantics based upon that. The main kinds of sentences that have been difficult for Montague and his followers are examined in some detail and accounted for in this alternative logic.

Chapter II: The Definition System

William H. Paxton
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975 - 14 October 1976, II-1 to II-52, October 1976

The Definition System consists of a metalanguage and a compiler. The meta-language, which uses augmented phrase structure (APS) rules in which a structure declaration gives the constituent possibilities and an associated procedure defines attributes and factors for phrases built by the rule, is designed to provide a way of integrating various types of knowledge while avoiding commitment to a particular overall control strategy. A major job of the Definition Compiler is to construct an internal representation of the definition for use by the Executive in processing sentences. Structure graphs are constructed by the Compiler from the phrase structure declarations, and LISP procedures are written and compiled to implement the rule procedures. The Compiler also builds an internal lexicon that includes special entries for 'multiwords.' Finally, lookahead information is computed and stored for categories and rules.

COMPUTATION

Decision Theory and Artificial Intelligence II: The Hungry Monkey

Jerome A. Feldman
Computer Science Department, University of Rochester, NY

Robert F. Sproull
Xerox Palo Alto Research Center

Cognitive Science 1: 158-192, April 1977

The utility function of decision theory can be used to reveal tradeoffs among competing strategies for achieving various goals, taking into account such factors as reliability, the complexity of steps in the strategy, and the value of the goal. The utility function on strategies can therefore be used as a guide when searching for good strategies. It can also be used to formulate solutions to the problems of how to acquire a world model, how much planning effort is worthwhile, and whether verification tests should be performed. These techniques are illustrated by application to the classic monkey and bananas problem.

A Note on Associative Processors for Data Management

Glen G. Langdon, Jr.
IBM Research Laboratory, San Jose, CA 95193

Research Report RJ 1941, 14 pp., February 28, 1977

Associative processors are examined from a technological and engineering point of view, and a design directed toward a band of single read/write heads per track on a rotating magnetic media recommended. Some alternatives to the design of comparators, garbage collection and domain extraction of the Relational Associate Processor (RAP) architecture are offered. Methods to store several relations are suggested.

COMPUTATION

Viewing Control Structures as Patterns of Passing Messages

Carl Hewitt
Artificial Intelligence Laboratory, MIT, Cambridge, MA 02139

Artificial Intelligence 8: 323-364, June 1977

Intelligence can be modeled as a society of communicating knowledge-based problem-solving experts. Each of these experts can in turn be viewed as a society that can be further decomposed in the same way until the primitive actors of the system are reached. We are concerned with the ways in which actor message passing can be used to understand control structures as patterns of passing messages in serial processing. Actors are defined by their behavior and they interact on a purely local level. To set up such a system one must: 1) decide what kinds of actors to have, 2) decide what kinds of messages each actor can process, and 3) decide what each actor is to do with its messages. Also discussed: PLASMA (PLAnner-like System Modeled on Actors), Event diagrams graphic notation for displaying relationships among events of actor computation.

Chapter XIII: The Deduction Component

Richard E. Fikes
Stanford Research Institute, Menlo Park, CA 94025

Speech Understanding Research. Final Technical Report, 15 October 1975-14 October 1976, XII-1 to XII-60, October 1976

When called on (by the response component) to provide the answer to a question the deduction component can, 1) retrieve information stored directly in the nets, 2) derive information using general information stored as theorems in the net, 3) call user supplied functions pointed to in the net that obtain information from knowledge sources other than the net (such as data files). The deduction component accepts as input a QVISTA (a vista being a partition of the network) containing the network translation of an English query and a KVISTA containing the knowledge base from which answers to the query are to be retrieved. Processing entails seeking a match in the KVISTA for the query pattern. A successful match produces a list containing a binding for each QVISTA element to a corresponding KVISTA element. After a bindings list is returned it can be repeatedly pulsed to find as many different answers to the query as desired.

Spontaneous Computation in Cognitive Models

Chuck Rieger
Department of Computer Science, University of Maryland, College Park, 20742

Cognitive Science 1: 315 - 354, July 1977

In spontaneous computation (SC) code runs spontaneously rather than on demand. A LISP-based SC system includes: complex trigger patterns, the organization of trees of trigger patterns, and higher level organization and control of SC via a "channel" to which *watchers* (triggers) or *servers* may be attched at *tap points*. A channel is the medium through which one LISP function calls another function. SC can be used in cognitive models to model nonalgorithmic inference, to "follow" characters in a story comprehension system, to act as subgoal protectors and plan optimizers in a problem solver. Also discussed: SC, context and frames; ideas related to partially triggered SCs and their theoretical applications as context-focusers and motivation-generators.

FORAL LP for DIAM II: FORAL with Light Pen - A Language Primer

Michael E. Senko
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6328, 25 pp., November 22, 1976

FORAL LP (FORAL with Light Pen) is a data base language that uniquely capitalizes on light pen interaction. In FORAL LP the user constructs transactions by touching nodes and arcs of a binary semantic network. Advantages of this two-dimensional approach are: 1) The user works in terms of real world entities and associations rather than records and files. Constraints intrinsic to the network reduce the possibility of non-meaningful programs. 2) The user enters text by light pen, removing the need for typing skill and speeding up transaction entry. 3) The user is provided with immediate graphic feedback on his transaction constructions FORAL, in all its forms, works best with a DIAM II binary semantic network. This network can, however, be specified in special case forms that represent IMS hierarchies or report generator flat files.

COMPUTATION: PROGRAMMING: LANGUAGES

An Overview of KRL, a Knowledge Representation Language

Daniel G. Bobrow
Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304

Terry Winograd
Stanford University

Cognitive Science 1: 3-46, January 1977

KRL (Knowledge Representation Language) is an attempt to integrate procedural knowledge with a broad base of declarative forms. The formalism for declarative knowledge is based on *structured conceptual objects* with associated *descriptions*. These objects form a network of *memory units* with several different sorts of linkages, each having well-specified implications for the retrieval process. Procedures can be associated directly with the internal structure of a conceptual object. This *procedural attachment* allows the steps for a particular operation to be determined by characteristics of the specific entities involved. The control structure of KRL provides for a priority-ordered multiprocess agenda with explicit (user-provided) strategies for scheduling and resource allocation. It provides *procedure directories* with operate along with *process frameworks* to allow procedural parameterization of the fundamental system processes for building, comparing, and retrieving memory structures. Future development of KRL will include integrating procedure definition with the descriptive formalism.

Abstraction and Verification in ALPHARD: Introduction to Language and Methodology

William A. Wulf
Carnegie-Mellon University

Ralph L. London
USC Information Science Institute, 4676 Admiralty Way, Marina del Rey, CA 90291

Mary Shaw
Carnegie-Mellon University

USC Information Sciences Institute paper ISI/RR-76-46, 47 pp., June 1976

A key concept in structured programming is *abstraction*: the retention of the essential properties of a object and the corollary neglect of inessential details. *Alphard* is a programming language which makes use of abstraction mechanisms called *forms* to achieve its goals of supporting the development of well-structured programs and the formal verification of these programs. The important property of Alphard is its ability to separate the use of an abstraction from the definition of its concrete representation. The verificiation technique exloits this separation and permits the implementation (the *form*) to be verified independently of the abstract program in which it is used. In order to capture the symbiotic interaction of the two goals the language description is interleaved with the presentation of a proof technique and discussion of programming methodology. Examples are given.

COMPUTATION: PROGRAMMING: LANGUAGES

IPG-2: Input Program Generator

Peter W. Cook
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RA 85, 20 pp., January 4, 1977

The IPG program generates FORTRAN input programs intended to avoid difficulties caused by the sensitivity of FORTRAN READ commands to exact character position and the difficulty of re-entering, in a run-time directed manner, parts or all of the input data. IPG generated input programs achieve this by two methods: 1) all data is read by a simple free-format input scanner and 2) a keyword based input system is generated as specified by the IPG input, which allows for initial entry or subsequent modification of data under the control of one or more user-specified key words The syntax for statements processed by IPG is given as are examples of the use both of IPG and programs by IPG. IPG uses the input programs of DI;DO (IBRRB RA 84, abstracted elsewhere on this fiche).

DI:DO- Free-Format Input/Output for FORTRAN Programs

Peter W. Cook
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RA 84, 16 pp., December 8, 1977

Though the input/output mechanisms of FORTRAN are self-consistent, they require attention to details of column count that make fexible input somewhat difficult. DI;DO is a set of programs implementing a free-format system for use with FORTRAN programs. The basic concept of DI;DO is to set aside a subset of meaningful characters as "field separators" which serve to separate one "field" of data from another and to provide means to classify and evaluate the content of the resulting "fields." DI;DO has been used, in somewhat modified form, in several programs and does provide means by which a FORTRAN programmer can write very flexible input systems.

COMPUTATION: PROGRAMMING: LANGUAGES

An Empirical Study of List Structure in Lisp

Douglas W. Clark
Carnegie-Mellon University

C. Cordell Green
Computer Science Department, Stanford University, Stanford, CA 94305

Communications of the ACM 20: 78-87, February 1977

Static measurements of the list structure of five large Lisp programs have been analyzed, revealing substantial regularity, or predictability, among pointers to atoms and especially among pointers to lists. Pointers to atoms are found to obey, roughly, Zipf's law, which governs word frequencies in natural languages; pointers to lists usually point to a location physically nearby in memory. The use of such regularities in the space-efficient representation of list structure is discussed. Linearization of lists, whereby successive cdrs (or cars) are placed in consecutive memory locations whenever possible, greatly strengthens the observed regularity of list structure. It is shown that under some reasonable assumptions the entropy or information content of a car-cdr pair in the programs measured is about 10 to 15 bits before linearization, and about 7 to 12 bits after.

COMPUTATION: INFORMATION STRUCTURES

A Correspondence Between Two Sorting Methods

W. H. Burge
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6395, 12 pp., February 13, 1977

A new way to construct binary search trees has been discovered by C. J. Stephenson (IBRRB RC 6298, abstracted elsewhere on this fiche) in which the new item that is inserted into the tree becomes the root of the resulting tree. This new method is compared with a previous method which involves two stages: 1) A forest is produced that has keys in order both down a tree and from left to right in the roots of each list of immediate subtrees of a tree. 2) A sorted list is produced from the forest by removing the root of the first tree, i.e. the smallest key, leaving forests that are merged into one. The same process is then repeated on the resulting forest. The number of comparisons needed by both methods for a permutation is the number of pairs of elements of the permutation that have no element between them whose value is between the values of the members of the pair.

COMPUTATION: INFORMATION STRUCTURES

A Method for Constructing Binary Search Trees by Making Insertions at the Root

C. J. Stephenson
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10599

Research Report RC 6298, 18 pp., November 22, 1976

It is possible to construct a binary search tree by inserting items at the root instead of adding them as leaves. When used for sorting the method has several desirable properties, including: a) fewer comparisons in the best case, b fewer comparisons in the worst case, c) a reduced variance, and d) good performance when the items are already nearly sorted or nearly reverse sorted. For applications in which the tree is searched for existing items as well as having new items added to it (e.g. in the construction of a symbol table), the tree can be made to exhibit stacklike behaviour so that the fewest comparisons are required to locate the most recently used items.

Three Simple Node List Algorithms

Henry S. Warren, Jr.
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6364, 11 pp., January 19, 1977

A "node list" of a directed graph is a sequence of nodes such that every simple path in the graph is a subsequence of the node list. This paper gives three relatively simple algorithms for computing node lists. Previous results are summarized, and areas for further work are suggested.

IAM; Inferential Abstract Modeling - An Approach to Design of Information Models for Large Shared Data Bases

Janis A. Bubenko, Jr.
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6343, 79 pp., January 4, 1977

A conceptual schema for a data base is designed by collecting and integrating 'local' user information requirements and analysis of information inference relationships. The procedure consists of the following phases: 1) collection and specification of information requirements, 2) entity classification, 3) specification of functional dependencies, 4) abstract object specification, integration and analysis, 5) implied information analysis, 6) derivability (precedence) analysis and 7) transformation to an 'external-name-based' model. The phases are performed iteratively in a controlled fashion. Underlying theoretical concepts are defined and arguments presented for the method's applicability. Appendices demonstrate the application of the IAM method to a reasonably complex practical case.

The Temporal Dimension in Information Modeling

Janis A. Bubenko, Jr.
IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6187, 23 pp., November 16, 1976

The majority of modeling approaches pay no explicit attention to the temporal dimension. The information model of a particular application is seen as a finite, varying set of information objects normally reflecting the current (last observed) state of a model of some real-world sys em. Administrative systems, however, are not static and for some purposes it is desirable to utilize time as an independent analytic variable, thus requiring that the system maintain a 'history' of its transactions. A conceptual framework is introduced in which time is treated in an unrestricted fashion and binary and n-ary relational modeling approaches, as well as a few approches which include temporality as a basic concept, are examined for this veiwpoint. It seems advisable that, when designing a conceptual schema, a time-unrestricted design level should precede the specification of a finite, time-restricted computational schema.

COMPUTATION: INFORMATION STRUCTURES

Direct-Access Storage of Data Structures

Arnold L. Rosenberg
Mathematical Sciences Department, IBM Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6036, 48 pp., June 8, 1976

Direct access to a data structure requires the existence of a *coordinate system* for the structure. This essay presents state of the theory treatments of such questions as: What do coordinate systems look like? How can they be used when devising storage mappings for a data structure or a family of structures? (Here extendible rectangular arrays are given extended treatment.) Where do coordinates come from? - discussed in terms of the data graphs model. A 38 item annotated bibliography is given.

The D-graph Model of Large Shared Data Bases: A Representation of Integrity Constraints and Views as Abstract Data Types

Herbert Weber
IBM Research Laboratory, San Jose, CA 95193

Research Report RJ 1875, 42 pp., Nov. 22, 1976

The D-graph model offers a uniform notation to describe basic data structures like domains and relations, integrity constraints and views. The basic entities in the model are objects, which are characterized by types. Type specification is used to define the composition of objects out of other objects of different types and manipulation rules for objects. The concept of abstract data types is employed to provide the encapsulation of data objects by all the operations applicable to these objects. The concept has been applied to model and implement the synchronization of concurrent accesses to shared resources in operating systems and for the design of programming languages which support structured and modular programming. It is shown to be suitable for modeling integrity constraints and views and for the manipulation restrictions imposed by constraints and views.

Independent Components of Relations

J. Rissanen
IBM Research Laboratory, San Jose, CA 95193

Research Report RJ 1899, 27 pp., Jan. 10, 1977

In a multi-attribute relation or, equivalently, multi-column table a certain collection of the projections can be shown to be independent in much the same way as the factors in a cartesian product or orthogonal components of a vector. A precise notion of independence for relations is defined and studied. The main result states that the operator which reconstructs the original relation from its independent components is the natural join, and that independent components split the full family of functional dependencies into corresponding component families. These give an easily checked criterion for independence.

Database Abstractions: Aggregation

John Miles Smith, and Diane C. P. Smith
Computer Science Department, University of Utah, Salt Lake City, 84112

Communictions of the ACM 20: 405-413, June 1977

Aggregation transforms a relationship between objects into a higher-level object and is important in conceptualizing the real world. An aggregate is a data type which, under certain criteria of "well-definedness," specifies aggregation abstractions. Relational databases defined as collections of aggregates are structured as a hierarchy of n-ary relations. To maintain well-definedness update operations on such databases must preserve two invariants. Well-defined relations are distinct from relations in third normal form. These notions are complementary and both are important in database design. A top-down methodology for database design is described which separates decisions concerning aggregate structure from decisions concerning key identification. Aggregate types, and other types which support real-world abstractions without introducing implementation detail, should be incorporated into programming languages.

DOCUMENTATION: ABSTRACTING & INDEXING

An Analysis of Indexing Strategies

George Markowsky
Computer Science Department, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6038, 33 pp., June 8, 1976

Using the theoretical framework introduced by Chandra and Strong various indexing strategies are analyzed with respect to speed. The Promotion Strategy always does at least as well as the M-tree Strategy; the M-tree Strategy always does at least as well as the Best Root Strategy. The regions in which the Binary Tree Strategy dominates or is dominated by each of the other three strategies are derived. Graphs of the behavior of these strategies are given as well as for the Square Root Strategy for small values of N and small B. A more exact formula for the Binary Tree Strategy is derived and shown to differ insignificantly from the old formula.

DOCUMENTATION: ABSTRACTING & INDEXING

Automatic Skimming: The 'LOUISA' system

A. Michiels, J. Mullenders, and J. Noel
Université de Liège

Bulletin of the Assciation for Literary and Linguistic Computing 5: 2 14, 1977

LOUISA (Linguistically Oriented Understanding and Indexing System for Abstracts) INPUT: English Abstracts in the field of information science. SYSTEM: 1) *Dictionary.* **Entry format: semantic zone and subzones. grammatical zone and subzones. 2)** *Disambiguation.* **Three methods based on, a) semantic tables, b) grammatical and semantic information, c) 'structural parallelism'. 3)** *Computer programmes.* **PL/1 list processing, including automatic rule generation. OUTPUT: descriptors. LATER: paraphrases, Translations, etc.**

DOCUMENTATION RETRIEVAL

Mutatis Mutandis I: Preliminary Remarks on the Adaptation of Wilks' Preference Semantics System for Use within a Data Base Interrogation System

Margaret King
Institut pour les Études Semantiques et Cognitives, Universite de Geneve

Working Paper 30, 29 pp., 1976

Question-Answering may be regarded as a species of translation: The system translates the user's query into a formal language, the description of the query in the formal language constituting a set of procedures to be carried out in order for the query to be answered. The proposed system can be divided into five stages, of which three are internal to the system and invisible to the user: 1) INPUT is transformed into, 2) SEMANTIC REPRESENTATION, which is in turn transformed into 3) FORMAL QUERY LAGUAGE used for 4) DATA BASE SEARCH, which then provides 5) OUTPUT. This paper is primarily concerned with the transitions between 1 and 2 and between 2 and 3. Wilks' Preference Semantics is the semantic representation while the formal query language is designed to query a relational data base (Codd).

TORUS: A Step Towards Bridging the Gap Between Data Bases and the Casual User

J. Mylopoulos, A. Borgida, P. Cohen, N. Roussopoulos, J. Tsotsos, and H. Wong
Department of Computer Science, University of Toronto, Canada

Information Systems 2: 49-64, 1976

TORUS (TORonto Understanding System) is a natural language system serving as a front end to a data base management system. A semantic network is employed to store knowledge about a data base of student files and is used to find the meaning of each input statement, to decide what action to take with respect to the data base, and to select information that must be output in response. Interaction with the data base management system (DBMS) operates at three levels: a) *The semantic level.* Commands operate on the data base attributes with no knowledge about how attributes are divided into relations. b) *The interfacr level.* Commands operate on one or more data base relations. c) *The DBMS level.* Commonds operate on individual data base relations. An overview of the system is given and the following topics are discussed: structure of the semantic network, understanding input sentences, using DBMS during dialogue, sentence generation, implementation, shortcomings and new directions.

Production Rules as a Representation for a Knowledge-Based Consultation Program

Randall Davis, and Bruce Buchanan
Computer Science Department, Stanford University, Stanford, CA 94305

Artificial Intelligence 8: 15-45, February 1977

The MYCIN system acts as a consultant on the task of selecting antibiotic therapy for bacteremia. MYCIN has 6 components: 1) consultation program, 2) explanation program, 3) question answering program, 4) knowledge acquisition program, 5) patient database, 6) knowledge database. Domain specific knowledge is stored in a set of 200 production rules, each with a premise (a Boolean combination of predicate functions on associative triplets) and an action (indicating conclusions to be drawn if the premise is satisfied). A tree of contexts is constructed dynamically from a fixed hierarchy as the consultation proceeds and rules are invoked in a backward unwinding scheme that produces a depth-first search of an AND/OR goal tree. The system is fast enough for real-time interaction and an informal study has been completed in which experts approved of 72% of MYCIN's recommendations.

TRANSLATION

Automatic translation - A Survey of Different Approaches

Bernard Vauquois
GETA, Batiment CETA, Domaine Universitaire, BP 53, 38401 Grenoble Cedex, France

Statistical Methods in Linguistics 1976: 127-135

First generation systems (such as the Georgetown system) have a strategy based on a catalogue of linguistic facts which are locally relevant for a given pair of languages considered for translation in one direction. Second generation systems start with an input sentence, produce a structural specification of it, map that specification into a structural specification in the target language and generate the output sentence from that. Much of the progress within the second generation has been made by deepening the level at which transfer from source to target language is made, with the most sophisticated systems making limited use of semantic criteria. The third generation systems, dominated by the AI approach, are mainly (if not exclusively) oriented toward the semantic interpretation of texts. We now have practical systems of the first and second generation, but no third generation systems are yet practical. There is no foreseeable prospect for fully automatic high quality translation so all practical systems demand man-machine interaction.

SOCIAL-BEHAVIORAL SCIENCE

Deductive Verbal Models of Organization

Fred Wenstop
Bedriftsokonomish Institutt, Fryfsjavein 33c, Oslo 8, Norway

International Jourral of Man-Machine Studies 8: 293-311, May 1976

The idea that loosely defined simulation models of organizational behavior can yield more significant information than conventional precisely defined ones is explored using NL as the medium. The variables are formulated verbally rather than mathematically. A generative grammar is presented which restricts the set of allowed linguistic values and relations, thus making it possible to formulate a semantic model based on fuzzy set theory. The semantic model can then be used to calculate the dynamic behavior of verbal models, making it possible to infer future behavior given a linguistically stated initial state. A model of the general causes and effects of the use of bureaucratic rules (taken from Alvin Gouldner's *Patterns of Industrial Democracy*) was implemented in APL.

SOCIAL-BEHAVIORAL SCIENCE: PSYCHOLOGY

Identifying the Relevant Aspects of a Problem Text

J. R. Hayes, D. A. Waterman, and C. S. Robinson
Carnegie-Mellon University, Pittsburgh, PA 15213

Cognitive Science 1: 297-313, July 1977

Forty-nine subjects judged the relevancy of sentence parts of a word problem (the Allsports problem). Patterns of subjects' judgments suggest three problem-solving heuristics: a SETS heuristic, a TIME heuristic, and a QUESTION heuristic. Presentation of the question before the problem tends to suppress the SETS and TIME heuristics. A computer program (ATTEND) which simulates subjects' behavior is described. It is context-sensitive in that it can change a relevance judgment upon the acquisition of further information. Average subject judgments and ATTEND judgments agree for 87% of the items.

SOCIAL-BEHAVIORAL SCIENCE: PSYCHOLOGY

Definite Descriptions and Semantic Memory

Andrew Ortony, and Richard C. Anderson
University of Illinois at Urbana-Champaign, Urbana 61801

Cognitive Science 1: 74-83, January 1977

Referential uses of names and attributive uses of descriptions are *direct*. Attributive uses of names and referential uses of descriptions are *indirect*. Subjects exposed to sentences containing direct and indirect uses of names and definite descriptions tended incorrectly to reject indirect uses and exhibited false alarms to sentences involving direct uses. This finding contrary to the predictions, as do those of Anderson and Bower, and Rumelhart and Norman. Models of semantic memory must incorporate distinct intensional and extensional representations to avoid semantic distortion.

SOCIAL-BEHAVIORAL SCIENCE: PSYCHOLOGY

Functional Clauses and Sentence Segmentation

John M. Carroll
Behavioral Sciences Group, Computer Science Department, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

Michael K. Tanenhaus
Psycholinguistics Program, Department of Psychology, Columbia Umiversity, NY 10027

IBM Research Report RC 6307, 29 pp., November 29, 1976

In two experiments subjects listened to a sentence containing a brief tone, then wrote out the sentence and marked the location of the tone. The experimental sentences were bi-clausal with the tone placed before or after the clause break. The initial clause was either *functionally complete* or *functionally incomplete*. Functionally complete clauses contain a complete set of fully specified grammatical relations, while functionally incomplete clauses do not. In Experiment I, tones were mislocated towards the clause break and the final word of the first clause significantly more often for functionally complete clauses. Experiment II replicates this finding holding deep and surface structure variables constant. The results indicate that functionally complete clauses are better segmentation units during sentence perception than functionally incomplete clauses. Purely structural theories of the units of sentence perception cannot account for this finding.

SOCIAL-BEHAVIORA SCIENCE: PSYCHOLOGY

Two Experiments on the Comprehensibility of Pause-Depleted Speech

Don Nix
Computer Sciences Department, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598

Research Report RC 6305, 26 pp., November 24, 1976

The experiments investigated both objective and subjective comprehensibility levels of accelerated speech that was produced by depleting pauses from pre-recorded speech (done under computer control with the Speech Filing System developed at Watson). The comprehensibility of this pause-depleted speech was compared to that of speech carefully read from a transcript at the same rate (Experiment I) and to speech extemporaneously generated at approximately half the pause-depleted rate (Experiment II). The pause-depleted speech was found to be at least as comprehensible as the other types, and in certain cases more comprehensible. This argues for the viability of an automatic process that can reduce the listening time by at least 50 percent without reducing the comprehensibility.

SOCIAL-BEHAVIORAL SCIENCE: PSYCHOLOGY

An Experimental Study of Writing, Dictating, and Speaking

John D. Gould
IBM Research Center, Yorktown Heights, NY 10598

Research Report RC 6186, 20 pp., September 3, 1976

Subjects were briefly trained in the use of dictation equipment and then measurements were made of their performance while dictating 16 business letters. Pause (planning) times and review times decreased relative to generate times. In the comparison experiments which followed the quality and efficiency of subjects' dictated documents (letters and one-page essays) were equal to those of subjects' written documents, even though subjects had just learned to dictate. Speaking, in which a recipient listens to rather than reads the authors' document, emerged as a potentially useful composition method for Offices of the Future.

SOCIAL-BEHAVIORAL SCIENCE: PSYCHOLOGY

A System for Primitive Natural Language Acquistion

Larry R. Harris
Department of Mathematics, Dartmouth College. Hanover NH 03755

International Journal of Man-Machine Studies : 153-206, 1977

We desire to have the robot that can walk about the room, store information about the state of the room and answer questions about the room. And we wish to use natural language to control the robot. The robot is built with innate capabilities for physical action and for information processing. The latter problem in broken into three phases. In the first the words of the language are correlated with concepts (initially, with the primitive concepts). The robot improves its communication capability with time but never claims to reach the end. It can operate in Phase 3 at any time and can re-enter Phases 1 and 2 to acquire new word meanings and grammatical constructions at any time. Improvement comes by the teacher discovering deficiencies during Phase 3 and returning the robot to Phases 1 or 2.

The Psychological Unreality of Semantic Representations

J. D. Fodor
Department of Linguistics, University of Connecticut, Storrs, 06268

J. A. Fodor, and M. F. Garrett
Department of Psychology, MIT, Cambridge, MA 02139

Linguistic Inquiry 6: 515-531, Fall 1975

Both generative and interpretive semantics assert the necessity for rules of eliminative definition. However, there is no convincing evidence for the psychological reality of such. Intuitive arguments can be adduced against the reality of eliminative definition and experimental evidence concerning reaction time to achieve a correct evaluation of sentences containing various types of negatives suggests that such a level is unreal. If our arguments are sound then it appears practicaly mandatory to assume that meaning postulates mediate whatever entailment relations between sentences turn upon their lexical content.

SOCIAL-BEHAVIORAL SCIENCE: PSYCHOLOGY: LEARNING-ACQUISITION

nduction of Augmented Transition Networks

John R. Anderson
Department of Psychology, Yale University, New Haven, CT 06520

Cognitive Science 1:125-157, April 1977

The LEARNMORE part of the LAS (Language Acquistion System) program takes as its inputs a sentence, a semantic network representation (HAM) of the sentence's meaning (taken to represent the output of a picture parsing routine), and an indication of the main proposition of the sentence. It then induces an ATN which acts as a map that enables it to go back and forth between sentence and meaning. It induces the word classes of the language, the rules of formation for sentences, and the rules mapping sentences onto meaning. The induced ATN can be used for both generation and comprehension. Critical to the performance of the program are assumptions that it makes about the relation between sentence structure and surface structure (the graph deformation condition), about when word classes may be formed and when ATN networks may be merged, and about the structure of noun phrases. These assumptions seem to be good heuristics which are largely true for natural languages although they would not be true for many nonnatural languages. Provided these assumptions are satisfied, LAS seems capable of learning any context-free language.

Computer Acquisition of Natural Language: A Review and Prospectus

J. McMaster, J. R. Sampson, and J. E. King
Department of Computing Science, University of Alberta, Edmonton, Canada

International Journal of Man-Machine Studies 8: 367-396, July 1976

After a brief general discussion of language acquisition, four computer NL systems with acquisition components are reviewed: Schwarcz 1967, Kelly 1967, Harris 1972, Block *et al.* 1975. Based in large part on what has been learned from these previous efforts a new Comprehensive Language Acquisition Program (CLAP) is proposed. The heart of CLAP is the development of its parsing and related components through the sequential activation of five strategies: segmentation and meaning association, linear ordering, structural generalization, conflict resolution, and using discourse. The first three are sufficiently well-defined for computer implementation using established AI methodologies.

SOCIAL-BEHAVIORAL SCIENCE: POLITICAL SCIENCE

Ideological Belief System Simulation

Jaime G. Carbonell, Jr.
Department of Computer Science, Yale University, New Haven, CT 06520

Research Report No. 111, 30 pp., May 1977

POLITICS is an automated political belief system simulator. Given a story about a political conflict and an ideology to use in interpreting the story, POLITICS generates a full story representation using the knowledge structures of Schank and Abelson (1977), predicts possible future events, makes suggestions about what should be done to rectify the situation, and answers a wide variety of questions. A subset of politics can function like Abelson's Goldwater machine (1965), but it solves most of the serious problems faced by that machine. An ideological belief system is represented as the attribution of a set of goal trees. Goal directed inferencing processes were developed to inte[illegible] with scripts and counterplanning strategies were investigated.

Bulletin of the Association for Literary and Linguistic Computing

Volume 5, No. 1, 1977

*These articles have been abstracted elsewhere on this fiche.

Spatial Reference and Natural-Language Machine Control

Norman K. Sondheimer
Department of Computer and Information Science, The Ohio State University, 2036 Neil Avenue Mall, Columbus, 43210

International Journal of Man-Machine Studies 8: 329 336, May 1976

Spatial references can be framed by orientational systems supplied by a large variety of sources having their structure established by different conventions, with anthropomorphic properties and the orientation with which people are familiar with an object being the two classes considered here. The systems in which comparisons are made are axis-like or path-like. The best method of dealing with ambiguity in spatial reference in NL machine-control systems is by restricting the syntactic and semantic structure, although this entails trade-offs between naturalness and expressiveness.

Systems Neuroscience

Jacqueline Metzler, Ed.
Center for Systems Neuroscience, University of Massachusetts at Amherst

Academic Press, x+271 pp., ISBN 0-12-481850-6, $12.00, 1977

CONTENTS

INTELLIGENT CAI: CURRENT PUBLICATIONS

BOLT BERANEK AND NEWMAN, INC.

50 Moulton Street
Cambridge, MA 02138

ARTIFICIAL INTELLIGENCE AND LEARNING STRATEGIES, John Seely Brown, Allan Collins, Gregory Harris, 40 pp., June 1977

THE GOAL STRUCTURE OF A SOCRATIC TUTOR, Albert L. Stevens, Allan Collins, 20 pp., BBN 3518, March 1977

A TUTORING AND STUDENT MODELING PARADIGM FOR GAMING ENVIRONMENTS, Richard R. Burton, John Seely Brosn, 11 pp., February 1976

DIAGNOSTIC MODELS FOR PROCEDURAL BUGS IN MATHEMATICS, John Seely Brown, Richard R. Burton, 44 pp., BBN 3669, August 1977

COMPUTERS IN A LEARNING SOCIETY, John Seely Brown, Ira Goldstein, 15 pp., October 1977

SEMANTIC GRAMMAR: AN ENGINEERING TECHNIQUE FOR CONSTRUCTING NATURAL LANGUAGE UNDERSTANDING SYSTEMS Richard R. Burton, 93 pp., BBN 3453, December 1976

INFERENCE IN TEXT UNDERSTANDING, Allan M. Collins, John Seely Brown, Kathy M. Larkin, 30 pp., 1978

A PARADIGMATIC EXAMPLE OF AN ARTIFICIALLY INTELLIGENT INSTRUCTIONAL SYSTEM, John Seely Brown, Richard C. Burton, 19 pp., June 1977

UNDERSTANDING HAND-PRINTED ALGEBRA FOR COMPUTER TUTORING, Stephen C. Purcell, 65 pp., February 1977

ASPECTS OF A THEORY FOR AUTOMATED STUDENT MODELLING, John Seely Brown, Richard R. Burton, Catherine Hausmann, Ira Goldstein, Bill Huggins, Mark Miller, 98 pp., BBN Report 3549, May 1977

REACTIVE LEARNING ENVIRONMENT FOR COMPUTER ASSISTED ELECTRONICS INSTRUCTION, John Seely Brown, Richard Rubenstein, Richard Burton, 99 pp., BBN 3314, October 1976

PROCESSES IN ACQUIRING KNOWLEDGE, Allan Collins, 55 pp., Jan 1976

SYSTEMATIC UNDERSTANDING: SYNTHESIS, ANALYSIS, AND CONTINGENT KNOWLEDGE IN SPECIALIZED UNDERSTANDING SYSTEMS, Robert J. Bobrow, John Seely Brown, 27 pp., 1975

STEPS TOWARD A THEORETICAL FOUNDATION FOR COMPLEX, KNOWLEDGE-BASED CAI, John Seely Brown, Richard Burton, Mark Miller, Johan deKleer, Stephen Purcell, Catherine Hausmann, Robert Bobrow, 138 pp., BBN 3135, August 1975

MULTIPLE REPRESENTATIONS OF KNOWLEDGE FOR TUTORIAL REASONING, John Seely Brown, Richard R. Burton, 39 pp., 1975

APPLICATIONS OF ARTIFICIAL INTELLIGENCE TECHNIQUES IN MAINTENANCE TRAINING, John Seely Brown, Robert J. Bobrow, 14 pp., Oct. 1975

USES OF ARTIFICIAL INTELLIGENCE AND ADVANCED COMPUTER TECHNOLOGY IN EDUCATION, John Seely Brown, 18 pp., 1977

A MODEL-DRIVEN QUESTION-ANSWERING SYSTEM FOR MIXED-INITIATIVE COMPUTER-ASSISTED INSTRUCTION, John Seely Brown, Richard R. Burton, Frank Zdybel, 9 pp., May 1973

REASONING FROM INCOMPLETE KNOWLEDGE, Allan Collins, Eleanor H. Warnock, Nelleke Aiello, Mark L. Miller, 33 pp., 1975

ANALYSIS AND SYNTHESIS OF TUTORIAL DIALOGUES, Allan Collins, Eleanor H. Warnock, Joseph J. Passafiume, 74 pp., BBN 2789, 1975

INTELLIGENT CAI, Allan Collins, Mario C. Grignetti. 33 pp., BBN 3181, October 1975