AD-A055 863 CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER --ETC F/G 5/7
AN ANNOTATED BIBLIOGRAPHY OF NATURAL LANGUAGE AND SPEECH UNDERS--ETC(U)
DEC 77 J R KENDER F44620-73-C-0074
UNCLASSIFIED AFOSR-TR-78-1093 NL

1 OF 1
ADA
055863

END
DATE
FILMED
8-78
DDC

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AFOSR-TR- 78 - 1093

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
AN ANNOTATED BIBLIOGRAPHY OF NATURAL LANGUAGE AND SPEECH UNDERSTANDING SYSTEMS

5. TYPE OF REPORT & PERIOD COVERED
Interim

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
John R. Kender

8. CONTRACT OR GRANT NUMBER(s)
F44620-73-C-0074

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Carnegie-Mellon University
Dept. of Computer Science
Pittsburgh, PA 15213

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61101E
A02466/7

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Projects Agency
1400 Wilson Blvd
Arlington, VA 22209

12. REPORT DATE
December 1977

13. NUMBER OF PAGES
57

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Air Force Office of Scientific Research (NM)
Bolling AFB, DC 20332

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Abstract: This annotated bibliography summarizes some 80 papers dealing with various aspects of natural language and speech understanding systems. Most detail a working system which "understands" "natural" input. Stress is on those issues at or above the level of syntax. Also included are several overviews and criticisms, usually in the form of comparative studies.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(2)

(6)

An Annotated Bibliography of Natural Language and Speech Understanding Systems.

(10) John R. Kender

(11) Dec[redacted]77

(12) 60p.

(9) Interim rept.,

DDC

JUN 30 1978

F

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa. 15213

(16) 2466

(18) AFOSR

(19) TR-78-1093

(17) 07

Abstract: This annotated bibliography summarizes some 80 papers dealing with various aspects of natural language and speech understanding systems. Most detail a working system which "understands" "natural" input. Stress is on those issues at or above the level of syntax. Also included are several overviews and criticisms, usually in the form of comparative studies.

Acknowledgments: The comments and criticism of Raj Reddy and Don Kosy are gratefully acknowledged.

This work was done while the author was supported by an IBM Graduate Fellowship. Partial support was provided by the Defense Advanced Research Projects Agency under contract F44620-73-C-0074 monitored by the Air Force Office of Scientific Research.

(15)

78 06 27 074 403081

1. Introduction

This annotated bibliography surveys but a small portion of the literature available in two related artificial intelligence research efforts. It encompasses many (but not all) natural language and speech understanding systems, and a sampling of the related research issues and criticisms.

In comparing speech and natural language systems, contrast rather than similarity appears to be the proper theme. Though all such systems aspire to the common goal of uncontrived man-machine communication, there seems to be only a few system characteristics that are universally shared. Much of the lack of concordance is due, in part, to the fields' youthfulness. A great deal of the research dates only after the ARPA speech study group report [Newell *et al.* 1971] and Winograd's natural language thesis [Winograd 1972]. Since these two landmark papers, much has been done, but little dogma has developed. As example, all books in this bibliography, but one, are solely collections of separate papers or are monographs on single systems. The exception, a textbook on natural language [*Charniak & Wilks*], begins with the curious disclaimer that the authors "disagree with each other on quite fundamental issues".

Neither youth nor disagreement, however, has deterred the production of a prodigious quantity of research. It is impossible to survey it all. Only working systems which understand natural input are included here. More specifically, this selectional criterion has three parts, each of which eliminates many otherwise interesting and appropriate papers. The stress on working systems rules out purely theoretical work, or those reports that detail proposed approaches or designs. Emphasis on understanding, as opposed to recognition, culls isolated word recognizers and purely syntactic parsers. A demand for natural input chiefly circumscribes various memory model efforts, which often presume a transducing "front end".

In addition, many other important and pertinent issues are considered to be merely peripheral to this paper. Among these are the volumes of work on the signal processing aspects of speech recognition, much related computational linguistics, the theories and programs for inferencing and theorem proving, and the essential concerns of knowledge acquisition and representation. What remains is a stress on those most communal aspects of complete, interactive understanding systems, namely those issues somewhere at or "above" the level of syntax. To motivate the abounding contrast found between speech and natural language systems, some criticism of the differing philosophies and methodologies of the two fields is also included. It is in these latter domains that the two disciplines seem to differ most sharply.

2. Natural Language Understanding Systems

[Siklossy *et al.*, 1972] L. Siklossy, and H. A. Simon, "Some Semantic Methods for Language Processing," *Simon & Siklossy*, pp. 44-66, 1972.

A review of several natural language understanding systems, and their relation to computational linguistics.

In contrast to linguists, who like to distinguish competence (abstract knowledge) from performance (actual use), the distinction disappears in computers. Basic attitudes in natural language understanding research: 1) There is a continuum of syntax and semantics. 2) There is a continuum of competence-performance. 3) There are many meanings of "meaning". 4) Natural language systems are complex: their performance cannot be derived; it must be tested against natural systems.

Three types of natural language understanding systems: hearer, speaker, and learner. In all work, it has been the resolution of ambiguity that has driven natural language. Reviews eleven programs, and notes the move to semantic grammars. Contrasts the two stages of linguistic processing (typically, mapping into deep structure followed by projection rules for search) with two stages of hearer programs (the structuring of the input followed by the performance of a task). In general, the more exact the parse, the more limited the input. For example, in Eliza, only an ordered list of important key words is matched to the input. Other programs are more brittle, though; all words must be known. This step often uses semantic knowledge. Task performance usually consists of information being placed in or "inferred" or "retrieved" from memory. Ambiguity can be handled by 1) use of context across sentences, 2) use of stored canonical forms, 3) "sensory information" (e.g. Coles' cathode ray tube screen input), and maybe 4) Quillian-like hierarchy distances.

Few speaker programs exists, due perhaps to paucity of world experiences. Learner programs (e.g. Siklossy's) usually have some semantics incorporated into the learned grammar. Siklossy's program builds patterns, mapping "pictures" into sentences: thus, the grammar is this mapping.

Summary: 1) Semantics can be used to disambiguate. 2) Meaning does not require phrase markers. 3) Grammar may be the rules transforming semantic structures into linguistic ones (but not necessarily using phrase markers).

[Wilks, 1974] Y. Wilks, "Natural Language Understanding Systems Within the AI Paradigm," Tech. Rep., Computer Science Dept., Stanford Univ., December, 1974.

A review, comparison, and criticism of five major natural language understanding systems; also addresses the relation of AI and computational linguistics.

Pithy, trenchant, and quotable: Cites "the profound role of fashion in artificial intelligence in its present prescientific phase." Cites "the fundamental role of metaphysical criticism in AI," that is, anyone who can speak feels entitled to criticize speaking programs. "Many of the principal researchers change their views on very fundamental questions between one paper and the next." "Criticism and comparisons are best drawn with a very broad brush and a light stroke."

Winograd's system presented. Grammar is a collection of small subprograms: procedures for imposing the desired syntactic structure. Heterarchical organization. Discussion of Shrdlu: 1) The linguistic system is highly conservative; its syntax and semantics distinction is unnecessary. 2) The semantics, tied to blocks world, is inextensible; blocks world is deductive and closed. One view is that Shrdlu is not about natural language, but about organization of goals. For example, the Planner code for "pick up" is not a sense of "pick up", but a case of its use. 3) Woods and Winograd both agree their formalisms are equivalent: both are grammar based deductive systems, operating in a question-answering environment, in a highly limited domain of discourse. Winograd's programs' Planner "suggestions" are like Woods' arc choices. However, Woods' position, that an assertion has no meaning if his system cannot establish its truth or falsity, is extreme.

Background issues in natural language: 1) Chomsky's insistence on competence models has isolated generative linguists from any effective test (i.e., performance). 2) "It may well turn out that the most appropriate form for plausible reasoning in order to understand is indeed non-deductive." 3) Procedural knowledge is not as perspicuous as declarative knowledge.

"Second generation" systems reviewed; characterized by the belief that understanding systems must be able to manipulate very complex linguistic objects. They are frame-like systems, which attempt to specify in advance how the world is structured.

1) Charniak: Understanding as pronoun resolution; based on partial (not necessarily true) information. Information in demons is highly specific (i.e., piggy banks, not containers). Charniak assumes "decoupling": that semantics and applications can be studied independently.

2) Colby: Parry is most used AI program. No syntax analysis; segmentation, then pattern match of segments, using 1700 rules. Patterns are tied directly to responses. Does Parry understand? "Many people on many occasions do seem to understand in the way that Parry does."

3) Simmons: Uses semantic nets of deep case relations, extended by paraphrase rules (e.g. "sell" and "buy" are considered forms of "transfer", etc.). Can be mapped into first order predicate calculus, for inductions.

4) Schank: Based on "dependency grammar" of Hays, has four conceptual categories (noun, verb, adjective, adverb), four cases, fourteen acts. Dictionary entries for verbs can be considered frames, seeking slot-filling items from context. Includes a theory of human mental acts: the representation of "John advised Mary" includes representations of Mary being pleased. Criticizes the stages of development of the system: ". . . consistent process of producing what was argued for in advance. . . . At each stage the representation has been claimed, in firm tones, to be the correct one." Some

problems: Word sense and prepositional ambiguity not addressed; primitives for only verbs and (possibly) nouns.

5) Wilks system: English-to-French translation task; is "reasonably robust"; based on preference. Templates are of the form: agent-action-object. Prepositions handled by templates of the form: dummy-action(preposition)-object. System never generates a deeper semantic representation than is necessary. Problems: 1) "Codings consisting entirely of primitives have a considerable amount of both vagueness and redundancy" (e.g. "hammer" and "mallet" indistinguishable). 2) Stability under large vocabulary questionable. Claims system is topologically similar to Schank; the heads of Wilks' formulae are like Schank's basic actions. However, Wilks' representation contains, by virtue of his word formulae, more information about what was anticipated.

Summary of the second generation systems. Two research styles apparent: finished product, and the developing system. Comparisons are hard to make due to a lack of precise theory in most systems. Compares them, however, in eight separate dimensions.

1) Levels of representation. Either language is represented by itself, or by primitives. Colby uses English directly and has enormous mapping problems. The ultimate defense of representation is perspicuity. Plausibility of Wilks' primitives defended by their similarities to the dictionary primitives of Webster. 2) Centrality. Specific or general knowledge: what leads to greater progress? 3) Phenomenological level. Pursuit of inference beyond "commonsense" is excessive (a comment aimed at Schank). 4) Decoupling. Can the parsing be considered separately from inference? (Charniak uses precoded structures, not natural language). Says no; parsing requires inference, as shown by the success of his and Schanks' semantic-based analyses.

5) Availability of surface structure. Appears sometimes necessary to include it, to preserve word sense (e.g. "nail", "screw", "peg" otherwise indistinguishable). 6) Application: perspicuity of procedures best in Winograd, worst in Schank and Charniak. Strongest objection is with case assignment of prepositions, which is not a mere "implementation issue" to be assumed. 7) Forward inference. As much as possible (Schank), or as little as necessary (Wilks)? Control problems occur with the former approach. 8) Justification of systems. Usually done on the following grounds: a) by the power of the inference system b) by the provision and formalization of knowledge c) by actual performance d) by psychological plausibility. Each system defines a natural language. Question is: How much is it like English?

Conclusion: What is needed is: good memory models, a theory for (multi-sentence) text, and a more sophisticated theory of causation. Also, error recovery from false expectations (as compared with the closed world where all analyses are immediately verifiable). Also, the ability to combine highly specific knowledge with general knowledge. Basic thrusts of AI-based natural language: 1) Theories must be programmable. 2) Theories must deal with language in a communicative context. 3) Theories must formalize and organize knowledge.

[Wilks, 1976] Y. Wilks, "Parsing English," *Charniak & Wilks*, pp. 89-100 & 155-184, 1976.

Two chapters of a textbook, based on the above paper, with addition of considerable detail on Wilks' system.

Natural language systems divided into content-motivated (e.g. Shrdlu) and structurally-motivated (Student). The former attempts to deal with the three type of natural language ambiguity: word sense, structural (e.g. prepositional) and referential (anaphora). The latter justifies its mechanisms by the function they serve in the problem domain. Shrdlu discussed, with some amplification of mechanism.

Additions to above paper: Parry is easier to extend than most programs, but fragile in that only paranoids are permitted to act the way Parry does. Adds the following to comparison section: 1) Levels of representation. Schank only has primitives for verbs. 4) Decoupling. "Parsing is essential, so it cannot be decoupled; it defines the significance of the semantic structure." 7) Forward inference: Reiger limits inferences simply by numerical cutoff.

Adds 9) Modularity. Winograd's is a three way heterarchy, while Schank and Wilks integrate syntax and semantics. 10) Scale of representation. "Representations must be justified in terms of some concrete problem that they solve." Large scale frames have so far only been justified by the "plot line hypothesis": that is, stories are only understood vis-a-vis a basic story type (a stance open to debate). 11) Real world procedures. The implicit hypothesis of much work: It is better to concentrate on the representation of human activities we know how to perform; one cannot understand language about activities that one cannot perform. But what is it that the non-performer does not understand?

Outlines Wilks' system: it converts a paragraph of text into a "semantic block". Processing steps: 1) Fragments sentences at key words, and words are expanded into their formulae. 2) Templates match formulae heads of fragments, followed by preferential expansion of all matched templates. The "most preferred" is chosen, where "preferred" means semantically most interconnected. 3) Inference, if necessary: paraplates rejoin the templates of a sentence. Usually, this means reattaching prepositional phrases, or resolving of some anaphora. Accomplished by semantic filters, one for each sense of the preposition, ordered according to preference. Pronoun resolution is not a well structured problem; both general and specific solutions are necessary. 4) If pronouns are unresolved by paraplates, then inference. Two types: action formulae (i.e., verbs) create new templates, by filling out all required grammar cases as if satisfied (e.g. if an action has a goal, assume it has been achieved). Secondly, templates can generate "common sense" templates; the shortest chain of linked templates is preferred. Thus, the system uses preference at template level, paraplate level, and inference level. Problem of control of inference not addresssed.

Criticizes Riesbeck: his system is based on expectation, and their satisfaction as soon as possible. Riesbeck claims that expectations are unordered. System is based on a "phenomenological fallacy" that assumes that since humans are never conscious of alternate parses, neither should be machines. Note that it is a surface-oriented parser: it is verbs seeking prepositions (not basic actions seeking cases). Riesbeck has no backup; cannot handle easily constructed counterexamples (e.g. "John gave Mary to the bridegroom.").

2.1. Overviews of Specific Issues

2.1.1 Syntax

[Bruce, 1973] B. Bruce, "Case Structure Systems," *IJCAI3*, pp. 364-371, 1973.

A synthesis of several case structure systems.

A formalization of case structure systems in first order logic, in order to formulate, analyze, and compare case structure systems. Includes the use of "case signals" (e.g. prepositions) and "case conditions" (e.g. semantic filters on noun "features"). Causation and purpose are considered "cases". Shows how Fillmore ("every language has a deep case structure" of at least six cases), Simmons (five cases), Schank (four "dependents"), and others can be modeled in his formalism. System is Chronos, an augmented transition network parser with flexible case structure (i.e., cases are user defined). Uses depth-first search for the satisfaction of a verb's cases: using the case conditions of the verb (procedural knowledge), it evaluates the probability that a noun phrase is a particular case. Admits of a rather haphazard interaction of various system components. Suggests that discourse analysis is easier if the case system is tailored to the situation.

2.1.2 Semantics

[Woods, 1975a] W. A. Woods, "What's in a Link: Foundations for Semantic Networks," *Bobrow & Collins*, pp. 35-82, 1975.

Mostly a detailed critique of the problems presently encountered with semantic nets.

Claims no theory of semantics exists yet. Also claims that canonical forms for English are unlikely (due, in part, to vague predicates: e.g. "uncle"); in any case, what is wanted is implications between concepts, not equivalence. Problems with semantic nets: Indefiniteness in regards to intention ("redness") versus extension ("a red thing"). Attribute-value pairs may have many kinds of things for value (i. e. numbers, relations, functions). Links stand for many types of relations: assertional versus structural (e.g. verbs versus their objects). N-ary relations not representable neatly. If relative clauses are represented by shared nodes, information is lost: what is subordinate? Indefinites ("a" versus "the", "need", "want") not distinguished from actual fact. The six possible quantifications of "every boy needs a dog", etc., are not easily distinguished.

[Simmons *et al.*, 1971] R. F. Simmons, and B. C. Bruce, "Some Relations Between Predicate Calculus and Semantic Net Representations of Discourse," *IJCAI2*, pp. 524-530, 1971.

Details on equivalence of semantic nets and the predicate calculus.

Shows (informally) that semantic nets and predicate calculus are similar, and that semantic nets are to be preferred for computational reasons. However, only shows the equivalence for a subset of semantic nets admitted to be inadequate for natural language understanding. Semantic nets are better for handling "vague" terms like "some".

2.1.3 Frames

[Minsky, 1975] M. Minsky, "A Framework for Representing Knowledge," *Schank & Nash-Webber*, pp. 118-130, 1975.

A summary of the longer frames paper stressing natural language issues.

A frame is a data structure for representing a stereotyped situation. The top nodes are fixed, but the lower levels ("slots") are weakly filled with default values; they can be replaced, but always subject to certain conditions on what can fill them. Different frames of a frame system share the same terminals. Recognition consists of the selection of frames (with respect to goals), and the filling in of slots with data. Claims, after a point, processing is serial with large symbols rather than parallel with much data. Generative grammars are to frame rules as transformational grammars are to frame system transformations. Any type of change can be modelled by before-after frame pairs. A frame also includes as part of its data the most serious anticipated problems associated with the stereotyped scenario they handle.

Frames are connected in a "similarity network". In the network a difference arc connects two frames together; the arc is labelled with the one difference between the frames. Thus, such similarity nets tend to cluster into "villages" centered around frame "capitols" from which their distance is small. Therefore, a stereotype is a capitol; that is, a central representative frame. Suggests that instead of trying to reduce problem space searches, should rather rerepresent the space.

2.2. Pattern Matching Systems

2.2.1 ELIZA

[Weizenbaum, 1966] J. Weizenbaum, "ELIZA--A Computer Program for the Study of Natural Language Communication Between Man and Machine, *Comm. ACM*, Vol. 9, No. 1, pp. 36-45, January, 1966.

A description of Eliza, and a warning disclaimer concerning "understanding".

Program is a keyword-based simulation of a Rogerian psychotherapist. Input sentences are transformed according to a rule associated with the keyword; handles single sentences only (rest omitted). Program is a simple driver, and a "script" of data (keywords, their rank, and their transformations). Pattern-matches on keywords of input; certain input phrases are carried over into output. Some transformations are mandatory (e.g. "I" -> "you"). Reassembly rules are used sequentially, then reused in course of conversation. Dynamically creates and stores extra transformations to be used when no keyword is present (e.g. "Earlier you said that . . .").

Domain was chosen since a psychotherapist is free to assume pose of knowing almost nothing. Success depends on much favorable interpretation by user: "Shows how easy it is to create and maintain the illusion of understanding." Needs a user model; presently is merely a "translating processor".

2.2.2 STUDENT

[Bobrow, 1968] D. G. Bobrow, "Natural Language Input for a Computer Problem-Solving System," *Minsky*, pp. 146-226, 1968.

A presentation of the algebra word problem solver.

Task is algebra story problems; written in LISP, with some added string processing functions. "Understanding" is taken to be exhibited by question-answering. Surveys several previous natural language programs. Claims to be the first implementation of "discourse analysis" (connected sentences).

Program uses "kernel sentences" and transformations on them. Assumes a naive user model: "What would I have meant if I had said that." Searches for instances of arithmetical operations; all the rest is considered "simple names" of variables. Solutions depend on resolving anaphora via pattern match, and via global knowledge (mathematical relations on the property lists of key word atoms). Processing consists of tagging of words by function (e.g. operator, or variable), and breaking sentences into kernel sentences by a primitive pattern match on "sentence formats" (i. e. connectives such as ", and"). Operator precedence rules then restructure the equations. One problem: Transformations are strictly order-dependent.

2.2.3 PARRY

[Colby, 1973] K. M. Colby, "Simulations of Belief Systems," *Schank & Colby*, pp. 251-286, 1973.

An overview of work on belief systems, featuring Parry and a summary of its validation using Turing indistinguishability tests.

Seeks belief systems which are "i-o equivalent", but can have different physical processes. Seeks "parallelism of behavior at some level". Human credibility does not follow strict mathematical axioms.

Outlines three predecessors of Parry. System 1: Neurotic belief. System altered the output of expressions of its beliefs, based on perceived internal conflicts. Abandoned, as belief base was thin, and there was no way to measure its neurosis. (Psychiatrists do not agree on "neurosis", but do agree on paranoia). System 2: Normal belief system. Domain is parent-child relations. Includes beliefs and "rules" (relations between beliefs and belief-classes). Data base sparsely related, though large; abandoned due to too much unconstrained search. System 3: Artificial belief systems. Credibility is assigned to new statements as a function of source, direct evidence, foundation beliefs, and consistency. But bogs down in search through a space of several thousand beliefs.

Parry is a simulated individual with a fixed set of malevolent delusions. Contains a context-free semantic grammar of "perceived intentions" of interviewer, which can be malevolent, benevolent, or neutral. Also has "flare" concepts which activate the delusional complex.

Input is classified by the grammar, and 1) internal values of affect (fear, anger, mistrust) are modified, and 2) output is produced (counterattack if angry, withdraw otherwise). Beliefs are here procedurally encoded as internal and external responses. Input is based on key words and rewrite rules: words are mapped into conceptual classes. Clauses and some other linguistic phenomena not handled. Hard part is input strategy: when to pursue current context. Heuristics are used; for example, if no new topic has been mentioned, look for an extension of previous concepts. Fear and anger are fluid, mistrust is not; simple mathematical formulae modify their values. Key word understanding simulates paranoids' ignoring of context when flare words occur. Also, paranoids are rigid, like the program. Uses canned responses of sentence length, some with variables that can be assigned to flare concepts.

Validation of model by means of Turing indistinguishability tests (reported below). Asserts the chief challenge is the widening of the scope of the model.

[Colby *et al.*, 1971] K. M. Colby, S. Weber, and F. D. Hilf, "Artificial Paranoia," *Artificial Intelligence*, Vol. 2, pp. 1-25, April, 1971.

A paper similar to the above, with some added detail on the semantics of the system.

Simple input is assumed; compound and complex sentences not handled well. Uses a keyword-based mapping of input into predications on an attribute of an object, or predications on a relation of the object to another object. "A combination of "you" or "your" with some form of the attribute, plus optionally another object or assisting concept will adequately convey the meaning." Data base is ordered so that object concepts occur before attribute concepts (distinguishes "your parents' residence" from "your residence"). Conceptual classes contain differing parts of speech ("work", "occupation") for ease in pattern matching. Uses a special scanner for specific grammar-based items: "I", "you", "me", metaverbs (e.g. "think"), positive or negative attitude tokens, passive forms, etc.

[Enea *et al.*, 1973] H. Enea, and K. M. Colby, "Idiolectic Language-Analysis for Understanding Doctor-Patient Dialogues," *IJCAI3*, pp. 278-284, 1973.

A paper detailing, with many examples, some of the specific production rules in Parry.

Cites the usual problem with dictionaries in semantic networks: adding a word or feature propagates strange side effects. Includes a long definition of "understanding", mostly relative to their own task. Input processor merges pattern matching with traditional parsers. Contains rewrite rules (productions) ordered by programmable precedence functions, but also contains "goal directed" rules which implement a context-sensitive grammar. Many interesting examples included. Rules are incrementally built up, by studying recorded dialogues.

[Colby *et al.*, 1974] K. M. Colby, and F. D. Hilf, "Multidimensional Evaluation of a Simulation of Paranoid Thought Processes," *Gregg*, pp. 287-293, 1974.

Details the application of several Turing indistinguishability tests to evaluate the accuracy of the paranoia model.

Conducted Turing-like indistinguishability tests with 41 psychologists and 67 computer scientists: both groups incapable of identifying which of two transcripts was machine or human. Forty psychologists rated similar transcript pairs along 12 dimensions. For example, Parry's language comprehension was poor, and mistrust was excessive, compared to a human paranoid transcript (though many characteristics were nearly equal). However, a version of Parry that output random replies was also evenly misjudged by 67 psychologists as human (except, as expected, "bizarreness" was higher). Conclusion: The Turing test is weak. Evaluations along several dimensions are much more important, as they can indicate what needs to be done with the model.

2.3. Microworlds

2.3.1 SHRDLU

[Winograd, 1973] T. Winograd, "A Procedural Model of Language Understanding," *Schank & Colby*, pp. 152-186, 1973.

An abridged version of the thesis describing and criticizing the Shrdlu system.

Microworld is a toy robot with arm that can manipulate blocks on a table. Can be commanded to manipulate, can be questioned about current and past states, can learn simple facts. A syntactic parser, some semantic routines, and a deductive system are the base; also includes a small response generator. Based on procedural knowledge; each of the three major parts is written in a different language. Has a large range of linguistic capabilities; for example, connectives, anaphora, etc. World model is symbolically encoded in triples of the form: "(category object property)" (e.g. "(is B1 block)"). The categories are used for ease in language generation. Planner assertions used to form a tree of subgoals for each action. A goal history is used to answer "how" and "why" questions. Claims "all language use [is] a way of activating procedures within the hearer."

Language is mapped into Planner procedures, which can execute, or add knowledge, or search for knowledge. Dictionary definitions contain "semantic markers" used in deduction (i.e., table is "inanimate", so can't be moved). Semantic markers are really calls to deductive routines. Some words ("one", "the") have elaborate semantic programs to test each possible word sense. Syntax written in Programmar, organized around syntactic units, each with an associated program. Based on "systemic grammar": each unit has features and functions. Integrated syntax and semantics; syntactic fragments are semantically verified. Parsing is left-right, with little backup necessary in practice.

List limits of approach: 1) Control flow is primarily syntactic; a heterarchy of syntax and semantics is more psychologically plausible. 2) Only a primitive use is made of context and of discourse rules.

2.3.2 Miller

[Miller, 1975] R. L. Miller, "An Adaptive Natural Language System that Listens, Asks, and Learns," *IJCAI4*, pp. 406-413, 1975.

A learning natural language program based on the microworld of tick-tack-toe.

Plays tick-tack-toe; uses contextual evidence, and asks questions of user, to determine the meaning of new term. Similar to speech acoustic error: linguistic errors are corrected using "higher level knowledge". Has fixed semantic concepts, but learns new descriptions of them. Carefully lists the program's limitations.

Levels of processing: local syntactic, semantic clustering, cluster expansion and connection (finds unknown words), contextual inference (possible only since the class of semantic primitives is very small). Claims methods are domain-independent. Syntax used as an aid in semantic clustering. Utilizes surface "frames" for each concept, containing the verb and its necessary verb cases. Meaning of unknown words are deduced by best match to frames available. Keeps a process history to answer "why" questions; the history acts as a semantic filter on new terms, also, by limiting interpretations. Clauses that are known with certainty help resolve uncertain ones in the same sentence, by establishing a board position, for example. Sufficient restraints in the resolution of unknown words can be coded because of the complete knowledge of the domain.

2.4. Augmented Transition Network Systems

[Woods, 1970] W. A. Woods, "Transition Network Grammars for Natural Language Analysis," *Comm. ACM*, Vol. 13, No. 10, pp. 591-606, October, 1970.

Describes augmented transition networks and compares them with other parsing algorithms.

A recursive transition network is a nondeterministic finite state machine, whose arcs may also be state names. It cannot handle sentences requiring agreement between nonadjacent parts; also does not show relations among transformed variants of a sentence (passive, interrogative, etc.). Augmented transition networks add transformational grammar aspects: partial phrases are built in registers, conditions and actions are allowed on arcs, registers also can have flags set. Five basic arcs are: input word category tests, other tests, a call for recursion, an end to recursion, and a jump to another state. Claims augmented transition networks are better than existing transformational algorithms, which are basically types of analysis by synthesis, implying exponential time. Conjectures that transformational grammars can be mechanically transformed into augmented transition networks.

Claims augmented transition networks are psychologically suggestive, and easily extensible, unlike other transformational systems. Claims they are better than existing transition networks, as they have an explicitly stated formal model, which is "natural" to the task of natural language. Lists advantages: 1) Perspicuity. 2) Generative power: constructions can have an unbounded number of constituents, and can also be used for language generation. 3) Efficiency of representations: common subparts of grammar are merged. 4) Efficiency of operation: can postpone decisions by keeping several identical analyses merged until they must diverge; also, backtracking is often accomplished simply by manipulating registers (no rescans necessary). 5) Flexibility for experimentation: incorporation of semantic and probability measures to find "most likely" parses, etc. Can be accelerated using the Earley algorithm, though no time bounds given.

[Kaplan, 1971] R. M. Kaplan, "Augmented Transition Networks as Psychological Models of Sentence Comprehension," *IJCAI2*, pp. 429-443, 1971.

A justification of augmented transition networks on the basis of their ability to model human perceptual strategies.

Purely syntactic (only) psychological phenomena are reviewed. Brief survey of psycholinguistic theory: deep structures are transformed into surface structures; the contrast of competence (model of grammaticality) versus performance (restricted by short term memory limits). Reviews augmented transition networks; they are comparable in power to a Turing machine.

Shows that augmented transition network complexity, when measured in number of transitions, corresponds to (intuitive) psychological complexity of sentence comprehension. However, part of this argument is critically dependent on the order in which arcs are searched on leaving a node. Claims, further, that the way designers of augmented transition networks gradually elaborate an augmented transition network models human linguistic development.

2.4.1 Woods

[Woods, 1973] W. A. Woods, "An Experimental Parsing System for Transition Network Grammars, *Rustin*, pp. 111-154, 1973.

A description of several experiments attempting to explore the power of augmented transition networks.

Claims augmented transition networks are "efficient transformational grammar parsers". Describes them (see above papers). Augmented transition networks use special "hold" registers for "left extrapositioned" sentence components (i.e., for interrogative sentences). Flexibility is shown by an example: changing the "forms" (phrase-building routines) on only three arcs was all that was necessary to change the output form from phrase-structure to dependency format. Augmented transition networks are nondeterministic. Thus arcs can be ordered by the probability of their successfully aiding the parse, or other heuristics; actual parsing, then, is neither breadth- nor depth-first.

Actual runtime experimental system incorporates backup facilities: a module for deciding, on failure, where to backup to and what to try next, using "weights" on suspended configurations. Several experiments are described: Well-formed substrings are saved; expensive. Selective modifier (i.e., prepositional phrase) placement tried: all possible contexts are found and semantically filtered for preference. Semantically guided parsing attempted: a parse is rejected if no interpretation exists. Conjunction resolution attempted, by exhaustively trying all possible parses; expensive. Reports on the performance of Lunar, an augmented transition network plus 150 semantic interpretation rules. It understands 80% of "real" input.

2.4.2 Heidorn

[Heidorn, 1975] G. Heidorn, "Augmented Phrase Structure Grammars," *Schank & Nash-Webber*, pp. 1-5, 1975.

Describes a parsing and generating scheme independent of, but much like, augmented transition networks.

Traditional phrase structure rules are augmented by conditions and structure building actions; the data structures allow consistent decoding and encoding of natural language. Word "records" are sets of attribute-value pairs (like LISP atoms). "Segment records" are used for segments of text, and are joined together via encoding rules. Encoding rules are productions: left side matches segments, and right side prescribes new segment records. Rules match on equality of record attribute values. Transformations consist of setting attributes in new records to either new values, or pointers to records. Thus, it can incorporate semantic relations via prestored database records.

Most analysis rules are semantic-based. Decoding is basically left-right, bottom-up. Expectation (backup) is handled by "rule instance records" which can be extended: breadth-first search. Decoding is also handled by production systems. But, some care is necessary to handle the ordering of productions. Present system has 300 records, 800 rules. Task is to construct a GPSS simulation program from an English description of a simple queueing problem. Claims it is similar to augmented transition networks.

2.4.3 Simmons

[Simmons, 1973] R. F. Simmons, "Semantic Networks: Their Computation and Use for Understanding English Sentences," *Schank & Colby*, pp. 63-113, 1973.

Outlines how syntactic nets, together with augmented transition networks, can be used for analysis, paraphrase, inference, and generation of output.

Hypothesizes "one central cognitive structure of semantic net form into which perceptions of speech, vision, action and feeling can map, and from which can be generated speech, physical action, hallucinations, feelings, and other thoughts." Based on deep case structure grammar of Fillmore; only five deep cases (causal actant, theme, locus, source, and goal). However, cases are not well defined. A verb's allowable case structures assign it to one of a small number of paradigms, according to how its cases can appear in sentences.

System has detailed rules for mapping suffixes, determiners, adverbs, etc., into attributed nodes. Semantic relations are required to be: deep case relations, attributive relations, modality relations, connectives, quantitative relations, set relations, and token substitution. Resulting nets can potentially be computational (through procedural encodings), logical calculus-like (since network relations are predicates), and conceptual (can be seem as a "deep structure").

The transformation from string to net is via an augmented transition network; however, the actions build up a semantic net, rather than phrase markers. English sentences are generated from the semantic nets as follows. Input is a semantic structure, together with a list of the desired constraints on modality (e.g. generate a question about the theme). After selecting a verb paradigm pattern based on the constraints, the pattern is input to an augmented transition network. Arcs generate output by computing actions based on the pattern and the input structure.

Answering questions: These "semantic" nets only abstract the syntactic relations from sentences. No attempt is made to abstract lexical equivalences (e.g. "lose", "defeat"). Thus, needs paraphrase rules in order to handle mapping of the cases from one verb to a synonym (e.g. "He was defeated", "He suffered defeat"). Some such rules need many conditions to allow the map; usually, these are words with many senses.

Paraphrase is accomplished using augmented transition networks: each paraphrase rule becomes a small augmented transition network. However, several other programs for pattern matching are also necessary (and are given in an appendix). Question-answering is done by matching the case tokens of the input with case tokens of stored assertions, or their paraphrases. A large database thus requires that each word list all the structures it appears in, as well as all the structures it can be paraphrased to. Notes that paraphrase can be recursive, and combinatorially endless.

Concluding comments: 1) Lexical content is also in the form of these "semantic" nets. 2) Semantic disambiguation is left unresolved. 3) These are really syntactic nets, which can be "paraphrased" into semantic nets, or into another language, or into procedures for action.

2.5. Semantic Primitive Systems

[Wilks, 1975a] Y. Wilks, "Primitives and Words," *Schank & Nash-Webber*, pp. 42-45, 1975.

An exposition on the philosophies of semantic primitives, and the methods for judging their effectiveness.

Schank' and Wilks' are the only systems with semantic primitives. Schank's is mixed: has primitives, plus English words. Claims such surface words should be allowed only if defined in primitives, perhaps "reentrantly", as in a dictionary. Adopts the new view that all primitives are a micro-language, that is, a natural language in themselves, with all the natural language problems. Thus, no justification on basis of size or composition of vocabulary is meaningful (as it would not be with English itself). Ultimate test of a primitive system is the performance. Compared his list of primitives with the SDC dictionary, which listed frequency of words used to define other words. Agreed approximately, up to the 80 or so primitives he used. One intuitive test of a good primitive choice: does it allow for interesting semantic generalizations?

2.5.1 Wilks

[Wilks, 1973a] Y. Wilks, "Understanding Without Proofs," *IJCAI3*, pp. 270-277, 1973.

An exposition of the analysis portion of a semantics-based English-to-French machine translation system.

System is based neither on linguistics, nor on theorem proving. Mechanical translation of English to French is a major test of semantic understanding. Justifies lack of a deductive system by claiming that it is false that "principles of logic play an essential role in our description of the world." Uses, instead, "commonsense" inference rules, which also are input to the system as English sentences.

System consists of "templates" bound together by "paraplates" and inference rules. All three data types are composed of about 60 "elements" (semantic primitives). Input words are replaced by "formulae", which are binary trees of semantic primitives. System uses preference rather than semantic restriction. Templates are basic actor-action-object triples which locate the "usual conversational" kernel messages implicit in a sentence (e.g. "He is good" is a form of the template "Man Be Kind"), and disambiguate word senses. Templates are stored in a BNF. "Only defense of choice of primitives is that a system actually works."

Analysis of kernel sentences proceeds as follows: Words are expanded into their stored formulae and formulae "heads" (prime primitives) are used to select a subset of templates. Templates are expanded by substituting, for their three elements, the formulae of all words in the sentence which contain those elements as their heads. "Density" of preference satisfactions (i.e., number of matching elements) within templates indicates proper parse. System makes no syntax-semantics distinction. It first fragments a paragraph of text by keywords into kernel sentences, expands them, resolves anaphora by "tie" routines which apply "paraplates" (semantic filters) between kernel sentence templates. Paraplates, which resolve prepositional modification, are ordered by semantic density of content: the most specific senses of the prepositions are tried first. Inference rules are tried only when paraplates fail to resolve anaphora. Inference is used to predict "missing" templates; shortest chain of missing connected templates is the best. Claims this methodology is superior to that of deductive programs, which work best on puzzles but not on natural input, the latter being based on preference semantics.

[Wilks, 1973b] Y. Wilks, "The Stanford Machine Translation Project," *Rustin*, pp. 243-290, 1973.

Outlines both the analysis and generation parts of a semantics-based English-to-French machine translation system; amplifies previous paper.

Opening remarks: Logical (predicate calculus) versus linguistic intermediate languages is not necessarily a conflict; the two representations reflect "two levels of human understanding". No strong syntax is necessary in the system. Uses semantic templates of form subject-verb-object, where some parts can be dummies (e.g. prepositions are considered "pseudoverbs", and have templates of the form dummy-preposition-object). Assumes a finite number of templates are adequate to represent "most" of "ordinary" English.

Fragmentation of input text is at punctuation, subjunction words, conjunction words, and prepositions. The final semantic representation consists of tied templates, rather than hierarchical structures. Does not claim universality of templates: "No inventory of templates can be proved to be correct."

The French output dictionary is a list of pairs: a semantic form coupled with a

French "stereotype", which contains implicit generation rules and actual French words. The generation rules test case conditions and sometimes, as in the case of objects of verbs, search for other form-stereotype pairs. The most specific stereotype is always preferred. The basic stereotype search is augmented by "concord" and "number" routines to handle the French inflections. Much procedural knowledge in stereotypes, however: "halt-points" in stereotypes prescan for special cases of word usage, to "handle linguistic idiosyncracy". In general, the more irregular the word, the more special information is in the stereotype, and less in any related modifying stereotypes.

2.5.2 Conceptual Dependency

[Schank, 1971] R. C. Schank, "Finding the Conceptual Content and Intention in an Utterance in Natural Language Conversation," *IJCAI2*, pp. 444-454, 1971.

An early version of conceptual dependency and its analysis of sentences.

Claims communication, not grammaticality, is key issue in natural language. Expectation is a major element in understanding. Lists six types of expectation: sentential, conceptual, contextual, conversational, individual memory, cultural memory. Outlines conceptual dependency theory, and its primitive conceptual acts. "Syntax is . . . a searching mechanism for already known semantic information." A primary problem is finding the verb; the system uses syntactic and conceptual heuristics. Major problem of analysis is "extracting the presupposed information implicit in an utterance." Analysis uses one stereotyped, general implication chain of verbs to help fill empty conceptual slots in the conceptualization being built.

[Schank, 1973] R. C. Schank, "Identification of Conceptualizations Underlying Natural Language," *Schank & Colby*, pp. 187-247, 1973.

A detailed presentation of the fundamental theories and structures of conceptual dependency.

Seeks a representation of meaning in an unambiguous, language-free manner. Syntax is not enough (e.g. "John's love of Mary was harmful." versus "John's can of beans was edible."). A natural language understanding system should never find more than one meaning at a time, as is the case with human linguistic expectation.

Sentences are mapped into conceptualizations consisting of nominals, acts, and modifiers. Acts are broken into primitives, to aid in paraphrasing. There exist basic conceptual rules for attaching various links and modifiers to the conceptual graph (tense, etc.). The conceptual level has its own syntax of permissible constructs and its own semantics of selectional filters.

The primitives of the theory include: relations of nominals (containment, location, possession), and conceptual cases (objective, recipient, directive, instrumental). The conceptual cases depend on specific acts, and are always assumed to be present, even if they have to be filled in by defaults. Conceptual relations include causality, time, and location. Notes that many verbs are descriptions of the relations of unknown actions (e.g. "prevent"), or the resulting states of such (e.g. "hurt"). Conceptual verbs (like "think") are handled by positing a "conscious processor" and "long term memory", to and from which conceptualizations are transferred. Physical actions have six basic primitive acts (e.g. move, ingest, etc.). Thus a total of 14 acts; inference rules are therefore not many in number. Examples of (hypothetical) parses by machine. Conceptual semantics eliminates troublesome parses; gives several examples of both syntactic and semantic ambiguity and similarity. Summary: The theory is based on the "moving about of ideas or physical objects."

[Schank, 1975b] R. C. Schank, "MARGIE, The Conceptual Approach to Language Processing, and Conceptual Dependency Theory," *Schank*, pp. 1-82, 1975.

The introduction to the three collected Margie theses; surveys conceptual dependency and its implementation.

The system inputs, paraphrases or infers, and outputs English sentences. Margie is a specific attempt to "model human psychological processes" through language-free meaning representations; language and thought are considered separable. Claims the best conceptual base form is the one which expresses the most information explicitly. Analysis is based on expectation. "Semantic rules are preference rules that select the best syntactic combinations." Claims that the meaning representations that make inference easiest are probably the best.

Reviews conceptual dependency theory. Theory also contains several primitive physical and conceptual states (e.g. "joy"). Many examples of conceptual dependency graphs given; admits many sticky issues are unresolved. On inference: "The real meaning of a primitive act consists of the inferences that are likely to be true when the act is present." Each act generates its own set of inferences, both forward (i.e., consequences), and backward (antecedents, though this is generally harder). Inference is simplified considerably by the use of semantic primitives.

2.5.2.1 MARGIE

[Schank *et al.*, 1973] R. C. Schank, N. Goldman, C. J. Rieger, and C. Riesbeck, "MARGIE: Memory, Analysis, Response Generation and Inference on English," *IJCAI3*, pp. 255-261, 1973.

A terse summary of the Margie system's three components.

System operates in either paraphrase or inference modes. Output module uses Simmons' program, with modifications. Reviews the conceptual dependency theory. Analysis uses syntax only when all else fails; processing is highly specific to verb. Analysis can be considered a sort of augmented transition network. Memory can generate five types of inference: normative, peripheral, causative, resultative, and predictive. The memory module's basis is causal chain expansion. "Inference molecules" are LISP procedures. Control loop is: inference, then repeat; inferencing is stopped by "interest" and "strength" parameters being too low. Generation: The issue of what to say has not been addressed. Generator has two steps: 1) Conceptual dependency is mapped into a case network. 2) Case network is mapped into surface forms. Uses discrimination nets to select the most specific verb to describe a conceptualization. Paraphrases are accomplished by using various other nearby nodes in the discrimination net, which require different case structures. Summary: Conceptual dependency is a canonical mapping which enables easy inferencing.

2.5.2.2 MARGIE: Analysis

[Riesbeck, 1975a] C. K. Riesbeck, "Conceptual Analysis," *Schank*, pp. 83-156, 1975.

A summary of the thesis on the analysis portion of Margie.

Introduction: Role of syntax is small. No clear division kept between linguistic and non-linguistic knowledge. Basic orientation: "The sentences understood are about human behavior." Analysis based on conceptual expectation. Admits of ad hoc approach: "The process of taking an example and expanding the vocabulary to handle it was the basic means of growth in the analyzer." Since the code is LISP, this usually had a procedural effect. Analyzer is a program monitor plus dictionary of about 60 verbs.

Overview: As a word is scanned, it adds requests to a request list. The request list is checked to see if any of the requests' conditions are satisfied. If so, then their associated programs are executed. Example: "John gave Mary a beating." Notes that each word can have several senses which must be distinguished. Analyzer has no backup; attempts to understand "while the sentence is being read." Claims it only has to worry about semantic ambiguity; semantics subsume syntactic ambiguity. Thus the analyzer only ever produces a single parse. Time is handled only in relative terms ("before", "after").

Overview of expectations and their associated programs ("actions"): They are much like augmented transition networks. Actions can modify almost everything; expectation (conditions) can be dependent on almost everything. Semantic features of a word are represented in conceptual dependency notation. Some syntax in the analyzer: three surface cases of subject, object, and recipient, determined merely by word order. No prepositions are ever considered, and noun pairs are not handled ("kitchen table"). Semantics of nouns are handled only superficially; stress is on verbs. Example: "give" has in its definition that the recipient is the first human noun phrase after the verb, and the object is the first physical object.

Multi-sentence analysis: admits of its inadequate treatment. Expectations are created between sentence pairs. The first sentence establishes preferences for the senses of a predefined class of verbs, in order to disambiguate them in the second (e.g. "John and Mary were racing. John beat Mary."). Notes that the only conjunction allowed is that between noun phrases.

[Riesbeck, 1975b] C. K. Riesbeck, "Computational Understanding," *Schank & Nash-Webber*, pp. 15-19, 1975.

A review and second look at the Margie analyzer, with future suggestions.

Claims comprehension is a memory process: basically simple mechanisms, with large data bases organized by key concepts. Sentence analysis is based on expectations. Admits to "no good control over set of expectations." Therefore, is planning extensions to program. One is labelling expectations as to purpose, in order to delete them when no longer valid. ("Purpose" is the case slot to be filled). Also, will incorporate dependency information between expectations: what case slots are prerequisite to an expectation.

2.5.2.3 MARGIE: *Inference*

> [Rieger, 1975] C. H. Rieger, "Conceptual Memory and Inference," *Schank*, pp. 157-288, 1975.

A summary of the thesis on the inference portion of Margie.

Introduction: All inferences are spontaneously generated. "This theory does not extend into the domain of deciding what is appropriate to say." Representation: Design criteria include language independence, and a psychological orientation. All concepts are stored in a fully-inverted data base for easy access. However, use of semantic "is-a" relations not well defined, mostly due to a lack of a taxonomy for nouns. Short term memory is simulated by "recency" tags. Beliefs and fact are distinguished by "truth" and "strength" tags. Inference chains are maintained together with "reason" and "offspring" lists. Real world knowledge is represented by patterns weighted by probability, which are matched against (e.g. "ingest person meat").

Inferences: Claims there is much subconscious, spontaneous (goal-less) inference to every stimulus; admits this psychology is "naive". The inferencing attempts to form "interesting" new relationships, in the manner of Quillian's expanding spheres. Contrasts his form of inference with 1) inference at question time, as in Planner data bases, 2) demons of Charniak, and 3) theorem provers, which have no analogue of his fuzzy logic. Inferences confirm, contradict, or augment existing knowledge.

Mainstream inferences: 16 types, only six of which are detailed. Inference needed since language tends to be as economic as possible. 1) Specification inferences. The filling in of obligatory conceptual cases with specific objects, mostly by problem-specific heuristic programs ("inference molecules"). Returns, also, a "reasons" lists, which allows interrogation of cause. 2) Causal and resultative inferences. Two types of inference: "cause" and "cancause", the latter being highly data-sensitive. Allows forward and backward causal chain expansion between two input conceptualizations, seeking a common intersection conceptualization. 3) Motivational inferences. Assumes "every real world action might have been volitional"; however, the motivation is inferred only if the actor could know about the results of his act. Special-purpose "normality molecules" rate the plausibility of generated motivations.

4) Functional inferences. When an object is wanted, its intended use is inferred. Unresolved are problems of knowing when to infer the more specific of functions (e.g. a newspaper used as a fly swatter). 5) Action prediction inferences. Inverse of

motivational inference, via molecules attached to each central act. Calls specification molecules to flesh out the predicted actions. "Illustrates how very sensitive all inference molecules must be to features of the objects involved in their inferences." 6) Utterance intention inferences. "I can't X" is really a request for X, etc. Open problem: how to handle the inferences that are derived from superfluous information (e.g. "Don't eat green gronks").

Inference-reference interaction: Problem is to disambiguate nouns. Process can involve arbitrarily many inferences, and the order of inferencing with respect to reference establishment varies. Solved by the creation of a temporary concept that is the intersection of the features of all possible referents. Inferencing now occurs, and the new inferenced information is checked against all candidates; the best match is the referent. Occasionally, normality molecules will aid inferencing in selecting the best candidate by making "most likely" inferences. This handles reference only locally, but claims the mechanism is general enough to work over several story lines.

2.5.2.4 MARGIE: Generation

[Goldman, 1975a] N. Goldman, "Conceptual Generation," *Schank*, pp. 289-371, 1975.

A summary of the thesis on the generator portion of Margie.

Introduction: Designed to be task and domain independent; used in Margie paraphrase and inference modes, and also to generate German output (machine translation). Overview: Word selection (mostly verb selection) is first step. Each verb has predicates ("defining characteristics") associated with it, which must be satisfied before the verb is chosen. Predicates may range over several conceptualizations in the input, or in the world model (e.g. "gave" versus "returned", "threaten" versus "promise"; depends on the "conceptual context"). Overriding philosophy: "A good generator will maximize the amount of structure encoded in the words it chooses."

Second step is syntax representation. Words are tied into syntactic networks of a weaker form than Simmons'; they have no conceptual significance. These networks determine the grammatical transformations (infinitive form, etc.) and word order. Each verb has associated with it the appropriate skeletal syntactic net. Augmented finite state transition networks produce the output.

Fine structure of the generator: Verb selection is via discrimination nets. Discrimination trees are binary trees with predicates at each node, which further specify the path to be taken to the terminal node specifying the output. The predicates check various "fields" of a conceptualization, by pattern matching and by inquiries into the world model. Some of these inquiries require deduction, which is not well handled. The discrimination trees are actually discrimination nets, as they have cycles to allow backup; they are hand crafted to prohibit looping. Fifteen nets, one for each major verb category. Admits of incompleteness of the nets, and of conceptual dependency itself.

Once verb is found, there is a pointer to a "concexicon" entry which holds the basic syntactic framework, plus programs for filling it. Scales of relative amounts are used for adjective selection. There are seven scales: health, joy, anger, excitation, physical state, size, and certainty; admits they are ad hoc. Language-specific functions are necessary to add language-specific information to the syntactic nets: for tense, determiners, possession, form (e.g. progressive tenses), and mood and voice (which are not actually handled).

An augmented finite state transition network generates output, based on Simmons' programs. Uses three separate "constructors" for verb strings, noun phrase strings, and sentences. Strictly a performance grammar, and admits to it being limited. Paraphrases achieved by using more general verbs, which are located higher in the discrimination tree.

[Goldman, 1975b] N. Goldman, "The Boundaries of Language Generation," *Schank & Nash-Webber*, pp. 84-87, 1975.

A review of some open problems in language generation.

Few have addressed the problem of "what constitutes a context requiring a natural language output." Most concern is with the representation in syntactic structures, in semantic nets, or in conceptual nets, or with the contextual effects on the utterance produced. Claims the assumption of single-sentence output is oversimplified. Reviews thesis work: representation is free of actual "words" and syntax, both of which must be reapplied. Notes that the conceptual nets have been designed to aid inference, not analysis or generation. Asserts that a model of the intended recipient's present state of understanding would aid generation greatly, but none exists yet.

2.5.3 Scripts

[Schank, 1975a] R. C. Schank, "The Structure of Episodes in Memory," *Bobrow & Collins*, pp. 237-272, 1975.

Outlines a theory of understanding based on causally linked actions.

Major focus: "How much information must be specified, at what level, in a meaning representation? To what extent can problems of inference be simplified by the choice of meaning representation?" Defends primitive acts: there is no right number of them, they overlap, they have "intuitive appeal only". Claims there are only four causal links between conceptualizations: result, enablement, initiation of thought, and reason for action. Although paragraph understanding is not implemented yet, asserts that understanding "is, in large part, the assigning of new input conceptualizations to causal sequences and in the inference of remembered conceptualizations which will allow for complete causal chains."

[Abelson, 1975a] R. P. Abelson, "Concepts for Representing Mundane Reality in Plans," *Bobrow & Collins*, pp. 273-309, 1975.

An outline of a system of primitives for expressing abstract state changes.

Concern is with belief systems; is conceptually close to Schank. Major parts of theory are: scripts (stereotyped action sequences), themes (related scripts), and dremes (attitudes toward themes). Cites the contrast between systems dealing with small worlds of complete knowledge, and systems dealing with big worlds of scattered knowledge. Favors the domain of political ideologues as a compromise. Theory primitives are nine "delt-acts", that is, acts which affect a change: for example, delt-proximity, delt-quality. Primitives are much higher level than Schank. Plans are sequences of desired state changes. Some problems remain: time passage is not formalized, and goals are not formalized.

[Schank *et al.*, 1975] R. C. Schank, and R. P. Abelson, "Scripts, Plans, and Knowledge," *IJCAI4*, pp. 151-157, 1975.

Presents a theory for understanding stereotyped and/or purposeful human activity.

Claims eventual limit to natural language understanding is the ability to characterize world knowledge. Defines understanding as "the fitting of new information into a previously organized view of the world". A script is a stereotyped sequence of actions in a context. There are many of them; some interact. Actions are linked by "causal chaining". The most interest, however, comes from deviations from the script. Script headers define the circumstances which fire the script. "What if" parts of the script handle obstacles or error. Reviews the program Sam: it instantiates a script, and makes inferences to complete causal chains.

Plans are a sequence of actions to realize a goal; they are infrequently used scripts. Composed of five primitive "deltacts". Each plan has a "plan box" associated with it which lists actions that achieve the goal; this list enables inferences. Pam, planned, handles plans. Claims "good forgetting is the key to remembering." Proposes to remember only a (non-script) event list, a goal list, a plan list, and a "weird list" of script deviations. Plans to "normalize" scenarios by replacing event lists and plan lists with pointers to "prototypes".

[Klein, 1975] S. Klein, "Meta-compiling Text Grammars as a Model for Human Behavior," *Schank & Nash-Webber*, pp. 94-98, 1975.

Outlines a very ambitious theory of human understanding, learning, and language.

Text grammars generate stories, somewhat like a script. Major concern, however, is with behavior transmission across generations. Wants to simulate the understanding, incorporation, and transmission of grammatical knowledge through simulated consciousnesses. Grammars are to be transmitted through example, inferred, and corrected through various interactions. Claims "it is the concepts of time and metacompiling that appear to be the fundamental aspects of human cognition." Example program creates many folk tales from a text grammar.

2.5.3.1 SAM

[Lehnert, 1975] W. Lehnert, "What Makes SAM Run? Script Based Techniques for Question Answering," *Schank & Nash-Webber*, pp. 59-64, 1975.

Details some principles and applications of stereotype-based understanding.

Sam is Script Applier Mechanism; answers questions about eating in restaurants. Explores some issues in question answering. The problem of the focus of the question (emphasis) is handled by the principle: When given a choice of focus, take variation over expectation. Questions are normally about what is variable in a script (e.g. Who was the specific actor?). Sam creates causal chains between input conceptualizations, according to the pattern of causal chains in the database script. Therefore, it can answer "what happened when" and "why" questions. The latter can be script-based or not, though only the script-based ones are well handled, by using both the temporal organization and goal sub-structure organization of the script. The script thus directs inference by focusing on variability. Claims system shows the power of episodic organization of knowledge, although it also incorporates semantic knowledge in the conceptual dependency framework.

2.6. Inferencing Systems

[Charniak, 1976] E. Charniak, "Inference and Knowledge," *Charniak & Wilks*, pp. 1-21 & 129-154, 1976.

Two chapters of a textbook, exploring the "narrower question of how knowledge is used to make inferences"; includes much of the second and third papers below.

Analyzes several systems of inference according to five aspects: 1) semantic representation used, 2) mechanism of inference triggering, 3) organization of programs and data, 4) inference mechanisms themselves, 5) content of the knowledge represented. First order predicate calculus and Planner are examined with respect to the above five criteria. A primary question is: When are inferences made, at question time or read time? Claims there is general agreement that some must be done at read time. Further question about read time inferences: How many should be made? Distinguishes "problem occasioned inferences" (to resolve anaphora), from all else ("keeping up" with a story). Claims non-problem occasioned inferences must be made, too.

Reviews his own thesis work, McDermott's Tople system, and Rieger's portion of Margie. Criticizes Reiger for his use of single sentences, simple actions ("hit"), and an unrestrained amount of inferences. The five criteria are applied to Charniak's thesis: 1) non-primitive semantics, 2) read-time inference triggering, 3) Planner procedures, 4) inference by demons, 5) no claims about content. Also applied to Reiger: 1) conceptual

dependency representation, 2) many read time inferences (16 types), 3) organized by inference and normality molecules (similar to Charniak's base routines and fact finders), 4) procedural inference, 5) no claims on content.

Frames, as applied to natural language, are reviewed. There are four basic types for language: syntactic, semantic, ones for stereotyped events, and ones for communication conventions. Claims Schank's scripts are frames.

[Abelson, 1975b] R. P. Abelson, "The Reasoner and the Inferencer Don't Talk Much to Each Other," *Schank & Nash-Webber*, pp. 183-187, 1975.

Some reflections on the philosophies of inference, and their problems.

Claims reasoning is formal, but inferencing is "commonsensical"; the two may be the same, though no one knows. A distinction is certainly true for humans; concrete information is used in favor of statistical, and the two types don't seem to combine readily. Gives interesting (human) examples, and asks if AI should simulate the dichotomy. Some methodological comments follow. A problem with AI is its diversity of problem contexts; claims there is a "tacit agreement that it is OK for everyone to define his own area." But by using his intention primitives, which represent state changes (nine "deltacts"), he can show similarities between the supermarket frames of Charniak ("fetching food") and the table top of Winograd ("fetching blocks"). However, claims that these primitives may be at too high a level, and not detailed enough, to actually use.

2.6.1 Charniak

[Charniak, 1973] E. Charniak, "Jack and Janet in Search of a Theory of Knowledge," *IJCAI3*, pp. 337-343, 1973.

A summary of some of his thesis work on inference.

Major concern is the organization of common sense knowledge to answer children's stories. Not strictly natural language understanding: sentences are hand-encoded. System flow: When a given topic is explicitly mentioned, its associated "base routines" set up "demons" which lie in wait for related events to occur in the following text. One problem: how to remove old demons (which may fire inappropriately, causing "misunderstanding", and which are inefficient). Inadequate solution is to remove them after N lines. System also includes "bookkeeping" routines to handle temporal relations, and "fact finders" to use standard inferencing (Planner) techniques. Some situations, especially those with both motives and results, are best handled by having demons call up other demons; each demon is a different abstraction of the situation. Piggy bank scenarios used as examples throughout.

[Charniak, 1975] E. Charniak, "Organization and Inference in a Frame-like System of Common Knowledge," *Schank & Nash-Webber*, pp. 46-55, 1975.

Presents a complete (theoretical) reworking of his theory of inference.

"Understanding a line of a story is to see it as instantiating one or more frame statements" of a frame. Gives several case analyses of frame problems using the scenario of shopping. A key problem: Given a statement, which frame statement is instantiated? Which of the frames themselves are active depends on "key concepts" "triggering" a frame; frame is then searched for a frame statement matching the story statement. Claims approach is better than demons: frames are more general, and can be used in multiple ways. For example, if frame statements are considered states to be achieved, they can be used to problem solve. Some additional problems: How many frames should there be, and how much is shared between them? Thus, in his example, the frame for shopping is augmented with a frame for a carry-cart, and common frame statements are shared via reference pointers.

The question of "read time" versus "question time" inferencing is not as serious as the problem of which inferences should be made. His answer: those inferences which serve to link frames (i.e., those that serve two purposes: e.g. completing a subframe, and filling in a frame statement in the main frame). Formally abandons the demon approach. One major problem with it was that topics had to be explicitly mentioned (not inferred). Claims that frames can handle the passage of time better, as they have room for the inclusion of "progress pointers" tracking the achievement of frame (script-like) events.

2.7. Other Interesting Systems

2.7.1 UNDERSTAND

[Hayes *et al.*, 1975] J. R. Hayes, and H. A. Simon, "Understanding Tasks Stated in Natural Language," *Reddy*, pp. 428-454, 1975.

A description of a general problem-solving natural language understanding system.

Task example is the "tea ceremony", an isomorph of the towers of Hanoi. One critical issue addressed is the construction of the problem space. System has two stages: language analysis, and problem construction. First part is retried if initial attempts to solve the problem fail. Based on Heuristic Compiler, a primarily semantics-based program. General rule: Rich semantics allows for weak syntax. An added "solver" part is similar to General Problem Solver. The front end provides it with 1) states, 2) operators, 3) a state differencing method, and 4) a connection table relating differences and applicable operators.

Stage I of the language analyzer maps text into deep structure via a case grammar. Uses Protocol Analysis System II. Three phases: a) Syntactic phase. Text is segmented by word class and punctuation; grammatical classes are assigned to groups of words; and integration rules match word-class patterns into syntactic units. b)

Semantic phase. Verbs are translated into relations, noun phrases into an assemblage of lists, sets, etc. c) Cross refencing phase. Anaphora and sentence joining, by matching: a pronoun is resolved if its verb and its suspected antecedent's verb are identical.

Stage II maps deep structure into problem representation. Whole text is scanned for participants and actions. Situations are found from those declarative sentences with indicated time lags. Operators are found from subjunctives and conditionals. Operations are matched against prestored prototypes (e.g. "transfer"), associated with which is procedural code for accomplishing its intent. Price for generality in prototypes is inefficiency. System design is evaluated using Moore and Newell's criteria. Only error handling rule: if interpretation is not clear, do not interpret at all.

2.7.2 SCHOLAR

[Carbonell *et al.*, 1973] J. R. Carbonell, and A. M. Collins, "Natural Semantics in Artificial Intelligence," *IJCAI3*, pp. 344-351, 1973.

A description of the Scholar program, and an investigation of various aspects of human semantic information and inference.

Major concern is the representation of information "in ways that are natural to people". Vehicle is Scholar program with "mixed initiative"; that is, is not merely a question answering program. Domain is computer aided instruction. Uses semantic nets, with hierarchical structure and "importance" ratings on the information content of nodes. Characterizes natural semantic information as 1) fuzzy (e.g. "large"), 2) incomplete, 3) contextual (handled in the system by checking the "importance ratings" of terms referenced by the speaker: nonexperts tend to stay at high-importance levels), 4) in an open world (the problem is when to say "I don't know" if knowledge can not be exhaustive), 5) with vague truthfulness ("often true"), and 6) vague quantification ("some"). Uncertainty is handled with "uncertainty ratings". Natural inferences are 1) deductive (using hierarchy relations), 2) negative (inferred through contradictions), 3) functional (i.e. procedural: e.g. using latitude to predict climate), and 4) inductive (not well understood).

2.7.3 SOPHIE

[Brown *et al.*, 1975] J. S. Brown, and R. R. Burton, "Multiple Representations of Knowledge for Tutorial Reasoning," *Bobrow & Collins*, pp. 311-349, 1975.

A description of the multiple knowledge sources, and problems, of the Sophie system.

Task domain is the computer-aided instruction of fault-finding in transistor circuits. Uses many types of knowledge: simulation, heuristic "procedural specialists" for various circuit components, and semantic nets (for static information). Input is parsed according to a semantic grammar. Grammar is also used to handle anaphora (semantic classes are used as filters on possible referents), for deletions, and for ellipsis. System is specifically designed for real time usage. An event history list also helps resolve ellipsis. System exploits the fact that inference in this domain can be achieved by (heuristic) simulation. Can even determine, by using resolution theorem proving, which requested circuit measurements would not add to the student's knowledge of the circuit problem.

2.7.4 ERMA

> [Clippinger, 1975] J. H. Clippinger, Jr., "Speaking with Many Tongues: Some Problems in Modeling Speakers of Actual Discourse," *Schank & Nash-Webber*, pp. 78-83, 1975.
>
> ***Describes a multiple knowledge source simulation of human language output.***

A human speaker monitors and regulates discourse as it is formed. Discourse is sensitive to speaker's goals, constraints, competence, and audience: "feedback regulated". Erma, written in Conniver, with five contexts (knowledge sources). They are Calvin (monitors acceptability of utterance), Machiavelli (monitors goal achievement), Cicero (models listener), Freud (models speaker), and the Realizer (generates the actual output). Data is structured in about 30 "concepts" (very small frames) which fire the modules through pattern matching. Uses a case-like grammar.

Claims: "Computational linguistics has yet to find its paradigm," since it was difficult to find a good framework in which to analyze some 200 actual dialogues. Calls for more empirical research in natural (not written) discourse.

2.8. Criticism

2.8.1 Criteria

> [Moore *et al.*, 1973] J. Moore, and A. Newell, "How Can Merlin Understand?," Tech. Rep., Computer Science Dept., Carnegie-Mellon Univ., November, 1973; also *Gregg*, pp. 201-252, 1974.
>
> ***Presents a list of eight design criteria with which understanding systems can be judged, and presents the Merlin system.***

Task of Merlin is the understanding of AI, through the understanding of AI programs. Definition of "understand": "S understands knowledge K if S uses K whenever appropriate." Notes that the presence of knowledge can be investigated directly in computer programs. "Appropriate" defined as "goal-serving". Understanding is difficult to test, as it requires a diversity of tasks. "Understanding may be partial both in extent and in immediacy." However, one possible test of understanding may be the understanding of natural language, which implies much understanding at large. Another test would be to satisfy a taxonomy of functional specifications that any understander is required to have; however, no such taxonomy exists.

In lieu of such a taxonomy, a taxonomy of design issues is proposed. Dimensions are: 1) Representation: with associated problems of scope, grain size, and multiple representations. 2) Action: output, and evocation of executable procedures. 3) Assimilation: input, and structuring of environment to existing representations. 4) Accommodation: the building of new internal structures, rather than the instantiation of old ones. 5) Directionality: goal directedness and "keep-going" ability. 6) Efficiency: including the possible problems of interpreters, general methods, and highly formal systems. 7) Error handling: including the "frame problem". 8) Depth of understanding: the "appropriateness" and ready access of knowledge. Three examples of understanding systems are judged according to the above criteria: predicate calculus theorem provers, Planner-like systems, and human beings (not well-analyzable yet).

Merlin itself uses beta structures to understand. A beta-structure is recursively defined as: "α: [β $\alpha 1$ $\alpha 2$ . . .]". That is, "α can be viewed as a β, if it is further specified according to $\alpha 1$, $\alpha 2$, etc.". Beta structures form a hierarchical knowledge net; however, the system does not make any deliberate generic-individual distinction. Structures can be mapped to one another. That is, beta-structure X can be viewed as a mapped version of beta-structure Y. This mapping is more powerful than general matching, since it can invoke the knowledge net hierarchy, and reinterpret any constituent beta structure. Merlin's use in problem solving: a problem is solved by attempting to see the current situation as a goal, and performing the necessary mapping. This imposes problem-solving mappings on the current situation's constituents.

2.8.2 Methodology

[Wilks, 1975b] Y. Wilks, "Methodology in AI and Natural Language Understanding," *Schank & Nash-Webber*, pp. 144-147, 1975.

Poses and answers three common objections to natural language understanding research.

Basic methodological disagreement: "Is there a science of language?" Three arguments: 1) Concerning theory and practice: "More theory is needed." Answered by: success in a task is the best test of a theory, not the theory's intuitive appeal. 2) Concerning AI and science: "Approximate success won't do." Answered by: AI is engineering. Easily constructed counterexamples do not, as in physics, overthrow what has been formalized. Due to nature of language, there is no boundary to natural language understanding, so no complete theory is possible. 3) Concerning where to start: "First need a theory of reasoning." Answered by: if so, then no one can understand anything unless he understands all.

[Mann, 1975] W. C. Mann, "Improving Methodology in Natural Language Processing," *Schank & Nash-Webber*, pp. 140-143, 1975.

Suggests many ways in which natural language understanding can be made more of a scientific endeavor.

Claims: "The style of research is the least flexible of precedents." Thus, natural language faces two problems: rigor and complexity. Parodies current research: select a phenomena, an input form, and an output form; code it; debug it on "examples of opportunity"; publish. "The activity is often treated as programming . . . rather than science." One problem is that the unit of production is the system, instead of the algorithm. Another is that the analyses usually center on only one of the processors in the intrinsically two-processor communication situation. Suggests the case analysis approach: data acquisition, phenomenon identification, case modeling, and model evaluation against the original data corpus.

2.8.3 Frontiers

[Woods, 1977] W. A. Woods, "A Personal View of Natural Language Understanding," *Waltz*, pp. 17-20, 1977.

An essay on what things are still required for a good natural language understanding system.

A good natural language understander must adequately handle: anaphora and ambiguity, quantification, adjectival and relative clauses, adverbs, conjunction and negation, time and tense, and paraphrases. Stresses the need for "practical theoretical solutions". One unresolved problem: The knowledge formulation must be flexible enough to allow eventual "closure", naturally. How to measure success and progress is difficult: there is no taxonomy of linguistic phenomena, and "perspicacity" of a system or a method is difficult to quantify.

3. Speech Understanding Systems

[Newell, 1975] A. Newell, "A Tutorial on Speech Understanding Systems," *Reddy*, pp. 3-54, 1975.

A review of various issues in speech understanding research.

Speech understanding as a research endeavor started about 1956. Its "dogmas": 1) The one performance criteria is understanding the message. 2) All sources of knowledge must be used. 3) The speech signal alone hasn't enough information. Outlines the structure of the task: "At present, there is no universal representation of meaning." Knowledge sources' knowledge is similar to linguistic "competence". Some mechanisms for converting knowledge to action: partial knowledge representations, combinatorial spaces, generative to analytic representation conversions, time versus frequency representations, matching algorithms, control of focus, multiple knowledge sources. Systems can be specified by ARPA's 19 dimensions, and by the system structure (hardware) and knowledge sources required.

Performance evaluation important: recall that the goal is a speech front end, not a system in itself. Can evaluate systems using benchmarks, operation research models, analysis of algorithms, null models (e.g. Dragon, Tech: relatively straight forward), optimal models (few exist), ablation studies (requires decomposability), analysis of variation, causality analysis (i.e. traditional debugging). Cites two tensions in the field: 1) interdisciplinarity, 2) general versus knowledge-specific mechanisms. Eventual scientific payoff includes: 1) understanding of human speech understanding, 2) formalization of influences on speech signal, 3) AI's first multiple knowledge source system, 4) disproof of statements that machines recognize with difficulty, 5) reinstrumentation of speech research. One practical payoff: can speak to computers.

[Reddy, 1976] D. R. Reddy, "Speech Recognition by Machine: A Review," *Proc. IEEE*, Vol. 64, No. 4, pp. 501-531, April, 1976.

Reviews several systems and their components, pointing out future directions for each.

All current speech understanding systems are "restricted speech understanding systems"; the restriction is the necessary use of task-specific information. Little common data, so comparisons between systems are difficult.

Connected speech recognition: difficult, since word junctures are not clear, and pronunciations vary with context; an "analyze and describe" paradigm is necessary, since the data is combinatorially large (no pattern recognition possible). In this class: Hearsay I, Dragon, Lincoln Labs' system, International Business Machines system. Knowledge is usually phonological rules, lexicon, and syntax. (The IBM system has *independent representations of language, phonology, and acoustic components, versus Dragon's uniform representation.)*

Speech understanding systems: Hearsay II; Bolt, Beranek, and Newman's; Stanford *Research* Institute-System Development Corporation's. Abandons traditional parsers' left-right scan of data. Symmetric acoustic knowledge sources required: "The role of knowledge sources is somewhat symmetric. They . . . predict or verify depending on context." Cites the need for easy addition or deletion of knowledge sources (for ablation studies, etc.). Some specific systems reviewed. SRI: parser controlled; language definition used to integrate knowledge sources; task is the management of a submarine data-base. BBN: developed using "incremental simulation"; augmented transition network is a basic component; task is a travel budget manager. Hearsay II: uses a "blackboard" model, and an hypothesize and test paradigm; task is news retrieval.

Task dependent knowledge required: vocabulary, syntax, semantics, pragmatics. Vocabulary is the primary source of restriction; confusability of words is key factor. Unstressed function words always a problem. Syntax: primarily a search reducer; restricts possible alternatives; measurable in terms of "branching factor". Most common is a network representation, including augmented transition networks; second is a Markov process model. Semantics, "rules and relationships associated with the meaning/ of symbols": another search space reducer. Semantic nets primary. Pragmatics, "conversation-dependent contextual knowledge" (for ellipsis and anaphora): handled by task dialogue models, basically tree structures. User model: predicts "modes" of interaction (query, clarification, etc).

System organization: usually best-first search or dynamic programming. Problems of focus of attention not well understood. Knowledge acquisition always difficult.

[Reddy *et al.*, 1974] D. R. Reddy, and A. Newell, "Knowledge and its Representation in a Speech Understanding System," *Gregg*, pp. 253-285, 1974.

A review of knowledge representation issues, using as an example Hearsay I under the voice-chess task.

Voice-chess was chosen since its syntax, semantics, and vocabulary are limited and well-defined. Some problems encountered in speech: 1) high data rate and large amounts of data, 2) errorful input, 3) real time response required. Uses separate knowledge sources and the "blackboard". Semantics module can rely on the fact that all a priori knowledge (chess rules) and all situational knowledge (the board state) are well defined. Even contains a primitive speaker model, in that the Tech chess-playing program ranks possible utterances for utility in the game. Syntax uses a context-free grammar, with "backward" "antiproductions" to predict from a given word permissible left and right word juxtapositions. Lexical knowledge has 31 words; uses knowledge of which syllables are stressed to help acoustic match. Presents a case study of "bishop to queen knight three".

Contrasts psychological active (motor) theories to passive (pattern recognition) theories; Hearsay is a blend. Claims pure analysis by synthesis is an unlikely model, due to efficiency considerations. Tabulates a taxonomy of types of knowledge necessary: at each level of speech processing, there are task, discourse, speaker, and analysis-dependent aspects of knowledge.

3.1. Overviews of Specific Issues

3.1.1 Organization and Control

[Reddy *et al.*, 1975] D. R. Reddy, and L. D. Erman, "Tutorial on System Organization for Speech Understanding," *Reddy*, pp. 457-480, 1975.

A review of some of the more practical aspects of speech understanding research.

Knowledge representation: In speech, one can exploit the well-defined linguistic levels; the units of knowledge in a higher level encompass more of the utterance. (Prosodics, however, is not a level). Error is ubiquitous; representations must be flexible. Semantic nets, augmented transition networks, production systems, and procedural embeddings possible.

Flow of control: hierarchy, heterarchy (sometimes based on incremental simulations), and blackboard have been used. Search is either by dynamic programming (conceptually, in parallel) or best-first search.

Research facilities required: real time input, quick tailoring of program parameters via "cliche" files, interactive debugging at a functional level, the handling of unplanned interrupts by user. Various types of performance analyses reviewed. Critical dimensions are accuracy, time, and space. Ablation experiments, "incremental improvement analysis" from studies of knowledge source interaction, algorithmic analyses are possible.

3.1.2 Syntax and Semantics

[Woods, 1975b] W. A. Woods, "Syntax, Semantics, and Speech," *Reddy*, pp. 345-400, 1975.

A review of some of the applications of computational linguistics to speech understanding systems.

Part I: Syntax. Reviews syntactic analysis schemes: phrase structure grammars (rewrite rules) and the Chomsky hierarchy of automata. Nondeterministic machines simulated using backtracking or parallelism: analysis is top-down, bottom-up, or mixed; predictive or not. However, in speech, phonological effects at beginning or end of sentences has bad effect on fixed order parsers. "Chart parsers" use word lattices to record well-formed substrings and their hierarchic dependencies; output an exhaustive list of all possible parse components (e.g. "Time flies like an arrow."). Earley's algorithm is a fast hybrid chart parser. However, a further problem in speech and natural language: languages are not context-free, and even the context-free part is complex.

Reviews use of transition network grammars: some arcs are labeled with phrase constituents ("push") allowing recursion and the merging of subparts of grammar. Transformational grammars are inefficient, and only one (perhaps) running computer program exists. Augmented transition networks have registers, and conditions and actions on their arcs; they are equivalent in power to Turing machines. In speech, augmented transition networks can be followed forward or backward to predict words, especially unstressed "function" words (prepositions, etc.).

Part II: Semantics ("the relation of symbols to meaning"). Reviews procedural semantics, as used in Lunar and Winograd. Lunar has a predicate calculus-like notation, directly translatable into Lisp procedures. Allows intentional (theorem proving) and extensional (execution against data base) reasoning. Semantic interpretation, via production rules, maps syntactic structures into procedures. Most such routines are verb-based.

Use of semantics in speech: Semantic selectional restrictions can be incorporated into the syntax to form "semantic grammars". But this fails to parse questions dealing with hypotheticality, or negation. Also fails for pronouns (no semantic classifications possible on the pronouns), and is inextensible. Would prefer something that also handles "default" word senses, and preferences. Cite use of semantics in speech for prediction as well as verification. Outlines the semantic nets of Quillian, where the meaning of X is considered the sum total of X's associated concepts. Such semantic associations can be used to predict; so can superset relations and inheritance of superset attributes.

3.1.3 The ARPA Projects

[Newell *et al.*, 1971] A. Newell, J. Barnett, J. W. Forgie, C. Green, D. Klatt, J. C. R. Licklider, J. Munson, R. Reddy, W. Woods, *Speech Understanding Systems: Final Report of a Study Group*, Carnegie-Mellon Univ., Pittsburgh, Pa., May, 1971.

Reports the philosophies and goals of the ARPA speech understanding projects.

Distinctive for its list of the 19 parameter values that a successful speech understanding system should have after the five-year effort. Basic viewpoint: Errors that count are errors in task accomplishment. Four task domains suggested 1) data base retrieval, 2) formatted data base entry ("voice key-punch"), 3) querying a computer system's status, 4) computer consultant (most ambitious of all). Each task is analyzed for possible control structures, and, at various speech levels (semantic, syntactic, lexical, etc.) for possible representations, knowledge and error sources, and problems. The 19 parameters are discussed in technical detail.

3.2. Connected Speech Recognition Systems

[Baker, 1975] J. K. Baker, "Stochastic Modeling for Automatic Speech Understanding," *Reddy*, pp. 521-542, 1975.

Reviews a specific technique's applicability to various levels of speech understanding.

These probability models can handle different types of uncertainty. For understanding, uses probabilistic model of a Markov process: matches observed acoustic vector Y to a sequence of random variables representing internal states of a Markov process X. Uses Bayes' theorem to evaluate the probability that Y(i) came from X(n), given probabilities that X produces Y(i). Markovian assumption of memorylessness simplifies computation: assumes that only the previous state (and not the entire preceding sequence) generates a given state. Examples of uses of this type of computation: many "low level" speech tasks. Outlines the Dragon system, in which linguistic, lexical, phonological, acoustic-phonetic, and semantic information are incorporated. All of Dragon's knowledge sources are probabilistic models of Markov processes, organized in hierarchies; dynamic programming is used to search for the best match. Thus, it analyzes all possible pronunciations of all possible sentences: still, time for utterance is linear.

3.2.1 DRAGON and HARPY

[Baker, 1974] J. K. Baker, "The DRAGON System--An Overview," *Erman*, pp. 22-26, 1974; also *Martin & Reddy*, pp. 24-29, 1975.

An overview of the Dragon system.

Model is: probabilistic function of a Markov process, plus dynamic programming to search the space. Recognition is linear in length of utterance; no combinatorial explosion. Stores a matrix for state-to-state transition probabilities. Signal match is via training, using Bayesian probabilities. Lexical knowledge is automatically compilable. Uses a very flat (non-hierarchic) network. Syntax and semantics are mixed in "task grammar" (chess is example). Training data is used for transition probabilities and signal match. Uses purely declarative knowledge, and straightforward search.

[Lowerre, 1976] B. T. Lowerre, "The Harpy Speech Recognition System," Ph.D. Thesis, Computer Science Dept., Carnegie-Mellon Univ., Pittsburgh, Pa., April, 1976.

Describes and criticizes Hearsay I and Dragon, as well as Harpy.

Harpy combines best features of Hearsay I and Dragon, though is most similar to latter. Hearsay I uses procedural knowledge, best-first search, and segmentation. Dragon uses Markov network with a priori probabilities, dynamic programming, and no segmentation. Harpy uses state transition network with data-dependent transition probabilities, heuristically modified dynamic programming, and segmentation. Also, simplifies the network by recognizing and coalescing common subnets, and includes word juncture phenomena in the network itself (Dragon had none).

Dragon system features include: probabilistic system of a markov process; state probabilities of the network updated every 10 ms. Network contains all syntactic and phonetic knowledge, represented by inter- and intra-state transition probabilities. Dynamic programming searches all paths, corresponding to all possible pronunciations of all possible sentences, to find best acoustic match. "Real action of the recognition process is due to the acoustic match probabilities".

Harpy: no interstate probabilities, just arcs (i.e. probabilities are one or zero) and intrastate transitions are dynamically calculated by reference to a table of minimum and maximum phoneme durations (and a heuristic threshold). Uses segmentation: performance is critically dependent on there being no missing segments; extra ones are easily handled by the network, however. Segmentation is based on linear predicive coefficients and several heuristic thresholds. Searching is sped up by only examining the (heuristically defined) "best" states of the network at any one utterance segment.

3.2.2 International Business Machines

[Jelinek, 1976] F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *Proc. IEEE*, Vol. 64, No.4, pp. 532-556, April, 1976.

Details the IBM series of speech recognition systems.

Systems are for speech recognition, not understanding. They model utterance production statistically, rather than through a semantic grammar. Phone-based stand-alone acoustic processor segments utterance; generates for each segment, through various estimates, the one best phone label and its start and end times. Speaker's phonetic performance is modeled on word base forms, and phonetic rules (e.g. coarticulations), plus rules that reflect the occasionally inaccurate idiosyncracies of the acoustic processor. Each word can be represented as a finite state machine, with the base form pronunciations and the phonetic rules providing the states and arcs. A Language Model is used to provide a priori probabilities for all words (the "New Raleigh Language", generated from a finite state grammar and 250 words).

One system approach: expand the language definition with word states, generating one very large finite state machine, and use the "Viterbi algorithm" (dynamic programming) to find best sequence of phones. Problem: This also gives the pronunciation of the string, which is unnecessary. An alternative: best-first search through the grammar ("stack algorithm of sequential decoding"). Best-first beats dynamic programming, probably because of a bad model of the acoustic processor (i.e. incomplete rules modeling its behavior).

[Bahl *et al.*, 1976] L. R. Bahl, J. K. Baker, P. S. Cohen, N. R. Dixon, F. Jelinek, R. L. Mercer, and H. F. Silverman, "Preliminary Results of the Performance of a System for the Automatic Recognition of Continuous Speech," *ICASSP*, pp. 425-429, 1976.

Reports on the performance of the above systems.

The system is an acoustic processor plus decoders; analysis is split is at phoneme level. Back end uses either a dynamic programming model given speaker and front end statistics, or a "stack decoder" which uses best-first search through grammar. Performance reported; also results of ablation studies: phonological rules removed. Also, tried various forms of speaker training: for example, by training front end only, and not back end.

3.3. Speech Understanding Systems

3.3.1 Hearsay I

[Reddy *et al.*, 1973b] D. R. Reddy, L. D. Erman, and R. B. Neely, "A Model and a System for Machine Recognition of Speech," *IEEE Trans. Audio and Electroaco.*, Vol AU-21, No. 3, pp. 229-238, June, 1973.

Presents an early version of Hearsay I.

Model: small set of cooperating independent processes, plus hypothesize and test paradigm. Parallel processes assumed necessary for real time response. Model is extensible and generalizable. Hearsay system modules include: speech input, speech output, task interface, and recognition subsystem (acoustics, syntax, semantics). Task is voice chess.

After a parametric level analysis and segmentation, the input is processed by 1) the acoustic recognizer (which has a hierarchy of increasingly accurate, but increasing costly tests), and/or 2) the syntactic recognizer (based on a grammar describing legal chess moves; "antiproductions" predict words to right or left of acoustically probable "islands"), and/or 3) the semantic recognizer (based on the chess-playing program Tech, which ranks legal moves by utility). Synchronization sequence is: 1) poll all, 2) "best" module hypothesizes, 3) the rest test. Voice chess appears to have a dominant semantics component.

The system is planned to have a "knowledge acquisition system" to dynamically update knowledge sources when parsing fails. Model is somewhat like analysis by synthesis except that individual words, not full utterances, are checked against input. Comments that highest level cognition is serial, but lowest (sensory) is parallel.

[Reddy *et al.*, 1973a] D. R. Reddy, L. D. Erman, R. D. Fennell, and R. B. Neely, "The Hearsay Speech Understanding System: An Example of the Recognition Process," *IJCAI3*, pp. 185-193, 1973.

An example of the performance of the above system.

Model is: diverse sources of knowledge, independent, cooperating, in parallel. Notes that errors at every stage of processing speech are possible (due to noise, lack of knowledge). System has three components: acoustic, syntactic, semantic. Voice chess is task; recognition paradigm is hypothesize and test. In actual performance, however, syntax or semantics only hypothesizes, and acoustics only tests. Implements a best-first search through the grammar. Recognition is word-based. An example is given; it includes processing errors (recovered from) in recognition. Knowledge sources reduce search space about a factor of five, at each stage of processing. Sources of knowledge also encompass what is known about speaker, environment, and transducer.

3.3.2 Hearsay II

[Erman *et al.*, 1975] L. D. Erman, and V. R. Lesser, "A Multi-Level Organization for Problem Solving Using Many, Diverse, Cooperating Sources of Knowledge," *IJCAI4*, pp. 483-490, 1975.

A generalized presentation of the blackboard approach to problem solving, based on Hearsay II; has an inclusive abstract.

In speech, much knowledge is required. However, knowledge sources are errorful and incomplete, due to deficiencies in theory, implementation (e.g. heuristic search), or data. Knowledge sources cooperate via a universal data base called "blackboard". This problem solving model uses the hypothesize and test paradigm.

Each knowledge source is independent, and knows of no others. Knowledge sources are derived from a "natural" decomposition of all task knowledge. Each knowledge source is fired by the pattern-matching of its precondition with the blackboard, much like an asynchronous production system. It changes the blackboard according to its knowledge.

The blackboard has many levels, one for each problem space decomposition level. Levels form a loose hierarchy, and imply a hierarchy of knowledge sources. Hearsay II blackboard has three dimensions: time in utterance, level of knowledge, and alternative hypothesis. Each hypothesize has attributes, among which are name, rating, "attention record" (processing time spent and/or recommended), and links to other hypotheses (forming an and-or graph).

Scheduling of knowledge sources is goal directed: if a solution path "stagnates", a new one is tried. The pattern matchers only look at new modifications to the data base. Can be easily parallelized. In Hearsay II, eight levels are linked by eleven knowledge sources plus some policy (e.g. scheduler) knowledge sources.

[Lesser *et al.*, 1974] V. R. Lesser, R. D. Fennell, L. D. Erman, and D. R. Reddy, "Organization of the Hearsay II Speech Understanding System," *Erman*, pp. 11-21, 1974; also *Martin & Reddy*, pp. 11-24, 1975.

Contains a critique of Hearsay I, and presents Hearsay II as an answer to some of its problems.

Task is news retrieval; system is designed for a multiprocessor. Uses "multiple diverse sources of knowledge". Knowledge sources are analyzed along four dimensions: function (poll, hypothesize, test), structure (independent), cooperation (through global data base, the "blackboard"), and attention focusing.

Hearsay I had a global data base of partial sentence hypotheses composed of words, with word and sentence ratings. Hearsay I problems: 1) processing in word units only, 2) lockstep control, 3) hypotheses were not linked to each other, 4) policy is hardwired. Hearsay II answers: 1) three-dimensional data base, with nodes at each linguistic level, utterance time, and alternative parse. 2) Preconditions for firing a knowledge source: data is directed by "matching prototypes" and is event-driven, like a production system. 3) And-or graphs between hypotheses propagate scores. 4) There is an independent policy module. Hearsay II levels are: conceptual, phrasal, lexical, syllabic, surface-phonemic, phonetic, segmental, parametric.

3.3.2.1 Organization and Control

[Hayes-Roth *et al.*, 1976a] F. Hayes-Roth, and V. R. Lesser, "Focus of Attention in a Distributed-Logic Speech Understanding System," *ICASSP*, pp. 416-420, 1976.

Discussion of the philosophy and implementation of control in Hearsay II.

The goal is minimization of knowledge source invocations. However, explicit control would destroy the flexibility of blackboard model. Basic approach: Each knowledge source action is summarized into a production: stimulus frame -> response frame. All decisions are based on these summaries.

Fundamental principles and mechanisms: 1) Best alternatives on blackboard are tried first. 2) More processing to knowledge source with more valid data. 3) More processing to knowledge source producing most significant changes. 4) Efficient knowledge sources favored. 5) Knowledge sources satisfying goals are preferred.

Variable called "state" at each time in utterance indicates the validity of hypotheses there; potential knowledge source contributions are measured against present "state". If no progress in an area of the utterance, then the knowledge source firing thresholds are lowered. Their output is also rated to be more credible than the uncertainty present in the area would normally warrant. Response to "state" can be breadth-first or depth-first. "Optimal strategy is not known." If "state" does not change for "a while", less desirable actions are tried in locations other than areas of high "state": prevents "cognitive fixedness".

Other knowledge sources ("policy modules") can modify the desirability ratings of various actions (response frames) effecting top-down, left-right, hybrid, etc., searches.

3.3.2.2 Syntax and Semantics

[Hayes-Roth *et al.*, 1976b] F. Hayes-Roth, and D. J. Mostow, "Syntax and Semantics in a Distributed Speech Understanding System," *ICASSP*, pp. 421-424, 1976.

Describes the design, construction, and execution of the syntax and semantics knowledge source of Hearsay II.

Addresses speech's "fundamental problem of uncertainty". Hearsay II uses no backtracking; rather, all alternatives are explicitly displayed. Uncertainty, combinatorial search, fuzzy (in the time domain) pattern matching, strong and weak inferences, and exploitations of partial information are addressed.

An input semantic grammar (declarative, not procedural) is converted automatically into productions of the form: preconditions -> response. Strength (hypothesis validity) associated with the production rules is inversely related to the size of the grammar class it covers. Four behavior rules for the knowledge source: recognition (creates phrases from words), prediction (outwards from "islands of plausibility"), respelling (gives components or alternatives for a predicted phrase), postdiction (post hoc support to hypotheses, i.e. a form of weak inference: predictions are not made, but reinforced if someone else makes them).

A recognition network is imposed as a filter on the blackboard for the detection of precondition satisfaction; it also records partial precondition information. Preconditions are governed by thresholds, which can vary over the utterance, allowing flexible attention focussing. All hypotheses are linked, and inherit "plausibility" ratings from their support.

3.3.3 Bolt, Beranek, and Newman

[Woods, 1974] W. A. Woods, "Motivation and Overview of BBN SPEECHLIS, An Experimental Prototype for Speech Understanding Research," *Erman*, pp. 1-10, 1974; also *Martin & Reddy*, pp. 2-10, 1975.

A presentation of an early form of Speechlis, featuring a description of the system-building technique of "incremental simulation".

Need for higher level knowledge in speech: human spectrogram reading experiments indicate that a 25% error rate can be reduced to 4% when syntactic and semantic information are allowed to the interpreters. System is based on Lunar system discourse models. The knowledge gathered through incremental simulation includes the fact that "function" words are missed by acoustics, and must be proposed by syntax.

Speechlis has six components: acoustic-phonetic, phonological, lexical, syntax, semantics, and pragmatics. Control consists of selecting best "theories" (hypotheses), and the establishment and execution of demonic "monitors". General control flow: First, segment lattice fills word lattice with words consisting of three or more phonemes.

Each such word is given to semantics and becomes a theory. The priority-governed best-first search ensues.

Results so far: Use of "fuzzy" (with respect to time) word matches reduces theories. Semantics can postulate semantic "clumps" (e.g. first person pronouns), also reducing theories. Pragmatics is in general too open-ended to use successfully, even though the speech signal has enough information to disambiguate otherwise confusing syntactic relations. Evaluations of the system are done with respect to human ("incremental") simulation.

> [Woods *et al.*, 1976] W. A. Woods, M. Bates, G. Brown, B. Bruce, J. W. Klovstad, and B. Nash-Webber, "Uses of Higher Level Knowledge in a Speech Understanding System: A Progress Report," *ICASSP*, pp. 438-441, 1976.
>
> *Overviews a later version of the BBN system.*

Travel budget manager is task. Data objects include 1) segment lattice (of phones, with probabilities, arranged chronologically), 2) theories (partial hypotheses of connected words), 3) monitors, notices, and events. Events are demons to watch for conditions in the word lattice; if conditions are met, notices are created and events (requests for further processing) are scheduled.

Based on a "pragmatic" grammar, which is topic-specific. A lexical retriever can predict the n best extension to islands, and control is by island-driving. First, the segment lattice is scanned left-right and right-left, to minimize word boundary effects. Then, the best words are found and put in the word lattice; each becomes a one word theory. The following is repeated until done: Syntax expands the "best" theory with words and/or word categories; lexical retrieval then replaces categories with words. "Fuzzy word matches" collect several related uncertain matches into one, if they are close in time. Island-driving from acoustically certain words is better than a strict left-right scan, as unusual phonological events occur at beginning and end of utterances.

3.3.3.1 Organization and Control

> [Rovner *et al.*, 1974] P. Rovner, B. Nash-Webber, and W. A. Woods, "Control Concepts in a Speech Understanding System," *Erman*, pp. 267-272, 1974; also *Martin & Reddy*, pp. 136-140, 1975.
>
> *Describes the design and performance of the control structures in the BBN system.*

Linguistic levels in the system are acoustic-phonetic, phonological, lexical, syntactic, semantic, and pragmatic. Data objects include the acoustic segment lattice, and the word lattice. Other data objects are theories (hypotheses concerning the original utterance), word monitors (which eventually cause condition-specific processing), and proposals (direct requests from one module to another). Evaluation of theories depends on acoustic match, duration information, syntactic and semantic scores, but almost no pragmatics. Control is started by the initial word lattice fill, and followed by evaluating and extending theories in order of priority. Some problems: theory of "thrashing" *(attention focussing) not good; incremental* simulation suggested to investigate it. Also, the theory of scoring utterance theories is inadequate.

3.3.3.2 Syntax

[Bates, 1974] M. Bates, "The Use of Syntax in a Speech Understanding System," *Erman*, pp. 226-233, 1974; also *Martin & Reddy*, pp. 112-117, 1975.

Outlines the syntax module in the BBN system, and describes some heuristics found necessary in its use.

Speech has "lexical ambiguity", that is, no clear word boundaries, no punctuation or capitalization clues. Also, small function words are unstressed, homonyms are confused ("see" versus "sea"), and word boundaries are lost ("tea meeting" versus "team eating").

Module uses an augmented transition network which hypothesizes basically top-down, (but can also operate bottom-up). An initial bottom-up pass of the acoustic modules constructs a "word lattice" with words of three phonemes or more. By "island-driving", the augmented transition network creates "monitors" on the lattice to look for hypothesized words. A problem: combinatorial explosion, as hypothesization is *breadth-first (all possible valid neighboring locations in the* augmented transition network are hypothesized). So, heuristics are used. One: scoring hypotheses and the use of threshold cutoff. Another: calling the semantics module for verification.

3.3.3.3 Semantics

[Nash-Webber, 1974] B. Nash-Webber, "Semantic Support for a Speech Understanding System," *Erman*, pp. 244-249, 1974; also *Martin & Reddy*, pp. 124-129, 1975.

Describes the semantics module of the BBN system, borrowed from Lunar.

Shows need for semantics: humans attain 90% intelligibility only when no more than two words have been excised from an eight word utterance. Uses "lexical semantics". Semantics most useful for "content" words (which are stressed). Word lattice is filled by the acoustic-phonetic, phonological, and lexical modules, initially with words with three or more phonemes only. Data structures include events, monitors, and theories (hypotheses).

Lunar semantic model: syntactic tree structure has restrictional templates; templates are referenced by their head noun or verb. Notes that semantic information is easier to retrieve in natural language systems. The semantic network contains multi-word nodes (allowing "horizontal" searches for related missing words), and relations between nodes (allowing "vertical" hypotheses). Relations in the network contain case frames, a type of semantic filter (e.g. the use of the word "ratio" requires that the two units be the same). Semantics hypothesizes new words, constructs theories, and evaluates using filters. Semantics also interacts with syntax, and translates the input sentence into the necessary procedures to execute the request. This latter illustrates one difference between recognition and understanding.

3.3.3.4 Pragmatics

[Bruce, 1975] B. Bruce, "Pragmatics in Speech Understanding," *IJCAI4*, pp. 461-467, 1975.

Outlines the task model and user model employed by the BBN system; has a rather natural language flavor.

Task is travel budget management task: user and task models employed. Intention of user (speaker): each speech act has presuppositions and desired outcomes. Presuppositions can be used as a filter on possible parses. Such an "intent" has preconditions, a case structure for its verbs, a list of desired outcomes, and pointers to examples (i.e. it is a type of frame). Examples of intents: "confirm data item", "ask again". Basic suppositions of sincerity necessary for success of user model.

Intents forecast future intents; expectation links form a "mode of interaction" (somewhat like a script). Modes have headers (preconditions) and a body of probabilistically linked intentions. Examples: "edit", "add" modes. Modes imply certain intents, which imply certain interpretations of speech. Thus, user and task model handle 1) expectations, 2) preference of parses, 3) actions to take (e.g. distinguishes between an "add data" intent and an implied "edit": "X is Y" adds Y to data base, unless data base has "X is Z".).

3.3.4 Stanford Research Institute

[Walker, 1974] D. E. Walker, "The SRI Speech Understanding System," *Erman*, pp. 32-37, 1974.

An overview of an early version of the SRI system.

System is guided and controlled by parser. Task is repairing a leaky faucet. Parser is a best-first searcher; uses a case grammar for verbs. Grammar allows anaphora ("it"). A microworld model is incorporated in the semantic network. A discourse model allows abbreviated responses, in the context of a discourse ("What bolt?" "That one."). Problems: Function words are unstressed, and words with liquids ("tool") are difficult.

3.3.4.1 Organization and Control

[Paxton *et al.*, 1975] W. H. Paxton, and A. E. Robinson, "System Integration and Control in a Speech Understanding System," Tech. Rep., Artificial Intelligence Center, Stanford Research Institute, September, 1975.

A description of the use of a "language definition" to unify and control the multi-module SRI speech understanding system.

Acoustic, syntactic, semantic and discourse knowledge sources integrated by "language definition": procedural knowledge consisting of word-based phrase composition rules (system is phrase-based). Each possible linguistic phrase (e.g. "verb phrase", "noun") has several attributes (as in Lisp, with values; all knowledge sources can contribute them) and several "factors" (validity scores, from any knowledge source). Each phrase, when built, is immediately assigned attributes and scores. New phrases match a pattern of a part of language definition, which fires and evaluates. Language definition also incorporates a discourse module (e.g. one attribute of a phrase is "interpretation", which is the phrase's referent found by the discourse module.). Six levels of factors: "very good" to "out".

Executive consists of a parse net and an associated task queue. Priorities are partly a function of phrase "value": the maximum possible score, over all sentences possibly containing the phrase, given a heuristic search over existing "contexts" (other active phrases). Also partly dependent on attention focussing, which is designed to keep activity from stagnating in one place, and is biased towards complete interpretations. Any partial results stored in parse net.

[Paxton, 1976a] W. H. Paxton, "A Framework for Language Understanding," Tech. Rep., Artificial Intelligence Center, Stanford Research Institute, June, 1976.

Sketches four critical dimensions in the design of the SRI speech understanding system.

Design issues: 1) System integration: both direct and indirect interactions are employed between the relatively large "tasks" (knowledge sources). System is phrase-based, and knowledge source procedures are triggered by phrase patterns. Phrase attributes are inherited by any larger, encompassing phrases. 2) Cooperation: handled by a parse net of terminal or nonterminal, complete or incomplete phrases; can be island-driven. 3) Evaluation: uses best-first acoustic choice. 4) Attention: focussed by the selection of "focus words" and its including phrases.

Claims this organization and these issues are applicable to natural language. Also claims that natural language is like speech in that 1) conjunction and comparatives create combinatorial explosion, and 2) ungrammaticality is like acoustic noise: some probabilistic method of choosing best interpretation is necessary.

3.3.4.2 Syntax

[Paxton, 1974] W. H. Paxton, "A Best-First Parser," *Erman*, pp. 218-225, 1974.

A description, and some performance analysis, of the SRI syntactic component.

Parser has four stages: syntactic (selects a legal grammatical class), lexical (selects a word), verification, and interparse cooperation. In verification, the priorities for a given parse are set using all other levels of knowledge. For example, semantic case frame agreement, word alignments in time (penalizes for gaps or overlap), acoustic match. Admits that setting priorities is highly empirical. In interparse cooperation, common subphrases are identified; old parts are integrated, along with their priorities, into new theories. Usually these subparts are noun phrases.

Relative performance analysis: parser performance is compared to a lower bound which is established by restraining it to the correct parse path; actual performance in best-first mode is three times this limit. A change to depth-first takes ten times lower limit. Also, there are studies with interparse cooperations toggled off and on.

3.3.4.3 Semantics

[Hendrix, 1975] G. G. Hendrix, "Expanding the Utility of Semantic Networks Through Partitioning," *IJCAI4*, pp. 115-121, 1975.

A theoretical paper on semantic nets, which is applied, in part, to the SRI system.

Main problem with semantic nets is quantification and hypotheticality. Solution: Arcs and nodes are separated into "spaces"; each arc or node is in exactly one such space. Each space has access only to itself and superset spaces: spaces thus can form lattices. Quantification (universal and its variants) is handled by quantifying individual elements within a semantic net subspace (the "form" of the propositions); quantified subspaces can be arbitrarily nested. This allows for the arbitrary mixing of universal and specific data. Partitioning also permits "want", "need", etc., to be distinguished from reality. Real versus hypothetical worlds discriminated; even discriminates hypothetical worlds from each other.

Used in SRI system to encode rules defining categories of objects (specifically, verb classes); this cuts down amount of information stored. Similar to use of "contexts" in some languages (say, Qlisp), but allows lattices, not just trees.

3.3.4.4 Discourse

[Deutsch, 1974] B. G. Deutsch, "The Structure of Task Oriented Dialogues," *Erman*, pp. 250-254, 1974.

An analysis of speech pragmatics, including task and user models; has a natural language flavor.

Problems of discourse analysis: "How does speaker decide what to include? How does the expression of new and old information differ?" Outlines some design issues of the pragmatics component of the SRI system.

There is much deictic ("pointing") information in the task environment, and much term definition. Hierarchy (actually lattice) of tasks implies a locality of reference within a subdialogue. Anaphora is resolved with respect to the task tree structure. Task hierarchy can be used in the anticipation of references. One unresolved problem: implicit closures of subdialogues (e.g. "I've got it" ends subtask).

3.3.4.5 *Evaluation*

[Paxton, 1976b] W. H. Paxton, "Experiments in Speech Understanding System Control," Tech. Rep., Artificial Intelligence Center, Stanford Research Institute, August, 1976.

Reports the results of an extensive array of performance experiments on the SRI speech understanding system.

Six experiments. 1) Test of acoustic mapper (word-based acoustic matcher). Finds function words generate most false alarms: 40 false alarms per good match ("hit"). This information used to simulate the mapper (and hypothetically better versions of it) in later experiments. 2) Language branching factor determined empirically, with and without acoustic restraints: usually three false alarms are better rated than the hit. 3) Two simple systems tested: dynamic programming on acoustics only, and a context-free grammar only. Both fail. 4) All cases of four binary control parameters tested on 60 utterances: island-drive versus left-right parse, breadth- or best-first acoustic checking of a set of proposed words, context checking, and selective focusing. Tested for accuracy and time. 5) Interword gaps and overlaps allowable in acoustic processor altered and found critical, due to word juncture phenomena. 6) Test of an increased vocabulary and improved acoustics (simulated by reducing false alarms). Result: 7% improvement in false alarm rate allows 50% bigger vocabulary. Summary: Acoustics is the bottleneck.

3.4. Criticism

[Neuberg, 1975] E. P. Neuberg, "Philosophies of Speech Recognition," *Reddy*, pp. 83-95, 1975.

A criticism of speech understanding research methodology.

Claims that success is due to increased computer power, and that research biases are simply reflections of various systems' "friendliness". Affirms that quantitative evaluations of techniques is difficult, and that the "scoring" of a parse is not well defined. Concerning prosodics: There is agreement to use it, in theory; but few do.

3.4.1 The ARPA Projects

[Medress, 1977] M. F. Medress, ed., "Speech Understanding Systems: Report of a Steering Committee," *SIGART Newsletter*, No. 62, pp. 4-8, April, 1977.

A short review of the achievements of the five year ARPA project in speech understanding research.

Success reported. One system, Harpy, achieved 97% semantic accuracy (91% word accuracy) on 1011 word vocabulary.

Three key aspects of the five year endeavor: 1) Multiple types of knowledge were brought to bear (syntax, semantics, coarticulation, phonology). 2) Many technical and scientific advances. 3) Interdisciplinary group. "A great deal is known, from the study of acoustics, phonetics, and linguistics, about the encoding of speech. . . . The sources of difficulty in understanding connected speech by machine are in the main rather well understood."

Reviews the five major research efforts, plus four minor ones, which resulted in four major systems. Harpy's success is due to the task-oriented grammar. Hearsay II had 91% semantic accuracy. Other two systems are less accurate, but use grammars that are less constrained. System-building techniques evolved. Linear predictive coefficients for the low end is now almost standard. Lists the 19 specifications of the original report, and gives Harpy's corresponding achievements of them.

[Klatt 1977] D. H. Klatt, *"Review of the ARPA Speech Understanding Project," J. Acoust. Soc. Am.*, Vol. 62, No. 6, pp. 1345-1366, December, 1977.

A review of the four completed systems, a summary of the scientific achievements of the project, and a forecast of possible future research.

Notes that the ARPA specifications did not require 1) tasks relevant to real-world problems, 2) "habitable" languages, 3) cost effectiveness. Success came from simplifying the problem by using syntactic and semantic constrains; thus the project was less successful in contributing to speech science. Harpy met or exceeded specifications.

The speech understanding problem: described by way of an example ("Did you hit it to Tom?") illustrating phonological difficulties, and a two-part paradigm of speech systems ("high end" and "low end"). The role of higher level knowledge is seen as that of constraint provision.

Speech understanding systems: four systems reviewed and discussed. Syntactic and semantic constraint can be measured by the average branching factor of the grammar.

SDC: Low end is syllable based; high end is best-first left-right scan of words. High end is sensitive to low-end errors. Discussion: Unclear why system failed. Possibly due to syntax's dependence on (usually unstressed) function words.

BBN: Low end produces a speech segment lattice, which can easily represent phonetic ambiguity. High end is island-driven, using an augmented transition network grammar with semantic constraints; search is thus best-first. System includes semantic procedures to produce an audio response. Discussion: Syntax is more general than other systems. Theoretical potentials unachieved, however; not enough optimization, perhaps.

Hearsay II: Organization is central blackboard with asynchronous knowledge sources (both low and high end). Word verification module is based on Harpy; system control is through island-driving. Discussion: Second best to Harpy, perhaps because of overall design. This included no absolute rejection of hypotheses, the optimization of components, and a grammar with the smallest branching factor.

Harpy: High end is an (acoustic) state network of all possible paths through a grammar, including word-juncture phenomena. Low end is based on linear predictive spectral match using Itakura metric. Search is heuristically-modified dynamic programming. Discussion: Success appears due to the network structure, the optimization of the network and the spectral templates, and strong syntactic restraints. "Harpy is essentially a verification strategy." The sparse network (i.e. grammar) appears more critical than low-end accuracy (only 40%). Notes that CMU had a variable branching-factor grammar, which was a powerful performance analysis aid.

Discussion and conclusions: Notes scientific advances in twelve broad categories. 1) System organization: Harpy's "beam search" and the Hearsay II blackboard. 2) Grammar design: CMU's variable branching-factor grammar; the use of branching factor, rather than vocabulary size, as a measure of complexity. 3) Control strategies: left-to-right is best only when function words are handled well. 4) Semantics and content: semantic grammars predominate.

5) Syntax: augmented transition networks are probably best for complex grammars. 6) Word verification: "Formal rules of considerable predictive power have been developed." 7) Acoustic-phonetic processing: Harpy shows that phonetic segmentation and labeling is not necessary. 8) Use of statistics: usually, it is impossible to get a large enough sample set.

9) Acoustic analysis: linear predictive coefficients or filter banks are both satisfactory. 10) Talker-normalization: Harpy's is automatic. 11) Response generation: which emphasizes understanding over recognition. 12) Contributions to speech science: includes the observation that some of the structures of speech understanding systems may be good models for human sentence comprehension.

A proposed future system: a Harpy-like low end, with an augmented transition network high end. Performance, however, would depend critically on "missing pieces" of speech science (e.g. a diphone dictionary). Cites the relationship of such a system to psychological models of speech perception. The proposed system makes four novel conjectures, including the human use of precompiled networks; but also leaves several questions unanswered.

Future research: Low-end: Key is the transforming of the phonetic identification problem into a spectral identification problem, as with Harpy. High-end: What is needed is realistic semantic constraints, and better human engineering. Other hard problems include increasing the grammar branching factor, distinguishing words that are more acoustically similar, and accurate function word recognition.

Four appendices are included that detail the SDC, BBN, Hearsay II, and Harpy systems.

4. References

[Abelson 1975a] R. P. Abelson, "Concepts for Representing Mundane Reality in Plans," *Bobrow & Collins*, pp. 273-309, 1975.

[Abelson 1975b] R. P. Abelson, "The Reasoner and the Inferencer Don't Talk Much to Each Other," *Schank & Nash-Webber*, pp. 183-187, 1975.

[Bahl *et al.* 1976] L. R. Bahl, J. K. Baker, P. S. Cohen, N. R. Dixon, F. Jelinek, R. L. Mercer, and H. F. Silverman, "Preliminary Results of the Performance of a System for the Automatic Recognition of Continuous Speech," *ICASSP*, pp. 425-429, 1976.

[Baker 1974] J. K. Baker, "The DRAGON System--An Overview," *Erman*, pp. 22-26, 1974; also *Martin & Reddy*, pp. 24-29, 1975.

[Baker 1975] J. K. Baker, "Stochastic Modeling for Automatic Speech Understanding," *Reddy*, pp. 521-542, 1975.

[Bates 1974] M. Bates, "The Use of Syntax in a Speech Understanding System," *Erman*, pp. 226-233, 1974; also *Martin & Reddy*, pp. 112-117, 1975.

[*Bobrow & Collins*] D. B. Bobrow, and A. Collins, eds., *Representation and Understanding*, New York, N. Y.: Academic Press, 1975.

[Bobrow 1968] D. G. Bobrow, "Natural Language Input for a Computer Problem-Solving System," *Minsky*, pp. 146-226, 1968.

[Brown *et al.* 1975] J. S. Brown, and R. R. Burton, "Multiple Representations of Knowledge for Tutorial Reasoning," *Bobrow & Collins*, pp. 311-349, 1975.

[Bruce 1973] B. Bruce, "Case Structure Systems," *IJCAI3*, pp. 364-371, 1973.

[Bruce 1975] B. Bruce, "Pragmatics in Speech Understanding," *IJCAI4*, pp. 461-467, 1975.

[Carbonell *et al.* 1973] J. R. Carbonell, and A. M. Collins, "Natural Semantics in Artificial Intelligence," *IJCAI3*, pp. 344-351, 1973.

[*Charniak & Wilks*] E. Charniak, and Y. Wilks, eds., *Computational Semantics*, Amsterdam: North-Holland, 1976.

[Charniak 1973] E. Charniak, "Jack and Janet in Search of a Theory of Knowledge," *IJCAI3*, pp. 337-343, 1973.

[Charniak 1975] E. Charniak, "Organization and Inference in a Frame-like System of Common Knowledge," *Schank & Nash-Webber*, pp. 46-55, 1975.

[Charniak 1976] E. Charniak, "Inference and Knowledge," *Charniak & Wilks*, pp. 1-21 & 129-154, 1976.

[Clippinger 1975] J. H. Clippinger, Jr., "Speaking with Many Tongues: Some Problems in Modeling Speakers of Actual Discourse," *Schank & Nash-Webber*, pp. 78-83, 1975.

[Colby 1973] K. M. Colby, "Simulations of Belief Systems," *Schank & Colby*, pp. 251-286, 1973.

[Colby *et al.* 1971] K. M. Colby, S. Weber, and F. D. Hilf, "Artificial Paranoia," *Artificial Intelligence*, Vol. 2, pp. 1-25, April, 1971.

[Colby *et al.* 1974] K. M. Colby, and F. D. Hilf, "Multidimensional Evaluation of a Simulation of Paranoid Thought Processes," *Gregg*, pp. 287-293, 1974.

[Deutsch 1974] B. G. Deutsch, "The Structure of Task Oriented Dialogues," *Erman*, pp. 250-254, 1974.

[Enea *et al.* 1973] H. Enea, and K. M. Colby, "Idiolectic Language-Analysis for Understanding Doctor-Patient Dialogues," *IJCAI3*, pp. 278-284, 1973.

[*Erman*] L. D. Erman, ed., *Contributed Papers of the IEEE Symposium on Speech Recognition*, Carnegie-Mellon Univ., Pittsburgh, Pa., April, 1974.

[Erman *et al.* 1975] L. D. Erman, and V. R. Lesser, "A Multi-Level Organization for Problem Solving Using Many, Diverse, Cooperating Sources of Knowledge," *IJCAI4*, pp. 483-490, 1975.

[Goldman 1975a] N. Goldman, "Conceptual Generation," *Schank*, pp. 289-371, 1975.

[Goldman 1975b] N. Goldman, "The Boundaries of Language Generation," *Schank & Nash-Webber*, pp. 84-87, 1975.

[*Gregg*] L. W. Gregg, ed., *Knowledge and Cognition*, Potomac, Md.: Lawrence Erlbaum Associates, 1974.

[Hayes *et al.* 1975] J. R. Hayes, and H. A. Simon, "Understanding Tasks Stated in Natural Language," *Reddy*, pp. 428-454, 1975.

[Hayes-Roth *et al.* 1976a] F. Hayes-Roth, and V. R. Lesser, "Focus of Attention in a Distributed-Logic Speech Understanding System," *ICASSP*, pp. 416-420, 1976.

[Hayes-Roth *et al.* 1976b] F. Hayes-Roth, and D. J. Mostow, "Syntax and Semantics in a Distributed Speech Understanding System," *ICASSP*, pp. 421-424, 1976.

[Heidorn 1975] G. Heidorn, "Augmented Phrase Structure Grammars," *Schank & Nash-Webber*, pp. 1-5, 1975.

[Hendrix 1975] G. G. Hendrix, "Expanding the Utility of Semantic Networks Through Partitioning," *IJCAI4*, pp. 115-121, 1975.

[*ICASSP*] ----------------, *Conference Record: 1976 International Conference on Acoustics, Speech, and Signal Processing*, Rome, N.Y.: Canterbury Press, 1976.

[*IJCAI2*] ---------------, *Advance Papers of the Second International Joint Conference on Artificial Intelligence*, British Computer Society, London, September, 1971.

[*IJCAI3*] ---------------, *Advance Papers of the Third International Joint Conference on Artificial Intelligence*, Stanford Research Institute, Menlo Park, Ca., August, 1973.

[*IJCAI4*] ---------------, *Advance Papers of the Fourth International Joint Conference on Artificial Intelligence*, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Ma., September, 1975.

[Jelinek 1976] F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *Proc. IEEE*, Vol. 64, No.4, pp. 532-556, April, 1976.

[Kaplan 1971] R. M. Kaplan, "Augmented Transition Networks as Psychological Models of Sentence Comprehension," *IJCAI2*, pp. 429-443, 1971.

[Klatt 1977] D. H. Klatt, "Review of the ARPA Speech Understanding Project," *J. Acoust. Soc. Am.*, Vol. 62, No. 6, pp. 1345-1366, December, 1977.

[Klein 1975] S. Klein, "Meta-compiling Text Grammars as a Model for Human Behavior," *Schank & Nash-Webber*, pp. 94-98, 1975.

[Lehnert 1975] W. Lehnert, "What Makes SAM Run? Script Based Techniques for Question Answering," *Schank & Nash-Webber*, pp. 59-64, 1975.

[Lesser *et al.* 1974] V. R. Lesser, R. D. Fennell, L. D. Erman, and D. R. Reddy, "Organization of the Hearsay II Speech Understanding System," *Erman*, pp. 11-21, 1974; also *Martin & Reddy*, pp. 11-24, 1975.

[Lowerre 1976] B. T. Lowerre, "The Harpy Speech Recognition System," Ph.D. Thesis, Computer Science Dept., Carnegie-Mellon Univ., Pittsburgh, Pa., April, 1976.

[Mann 1975] W. C. Mann, "Improving Methodology in Natural Language Processing," *Schank & Nash-Webber*, pp. 140-147, 1975.

[*Martin & Reddy*] T. B. Martin, and D. R. Reddy, eds., "Special Issue on IEEE Symposium on Speech Recognition," *IEEE Trans. Aco., Speech, and Sig. Proc.*, Vol. ASSP-23, No. 1, February, 1975.

[Medress 1977] M. F. Medress, ed., "Speech Understanding Systems: Report of a Steering Committee," *SIGART Newsletter*, No. 62, pp. 4-8, April, 1977.

[Miller 1975] R. L. Miller, "An Adaptive Natural Language System that Listens, Asks, and Learns," *IJCAI4*, pp. 406-413, 1975.

[*Minsky*] M. Minsky, ed., *Semantic Information Processing*, Cambridge, Ma.: MIT Press, 1968.

[Minsky 1975] M. Minsky, "A Framework for Representing Knowledge," *Schank & Nash-Webber*, pp. 118-130, 1975.

[Moore *et al.* 1973] J. Moore, and A. Newell, "How Can Merlin Understand?," Tech. Rep., Computer Science Dept., Carnegie-Mellon Univ., November, 1973; also *Gregg*, pp. 201-252, 1974.

[Nash-Webber 1974] B. Nash-Webber, "Semantic Support for a Speech Understanding System," *Erman*, pp. 244-249, 1974; also *Martin & Reddy*, pp. 124-129, 1975.

[Neuberg 1975] E. P. Neuberg, "Philosophies of Speech Recognition," *Reddy*, pp. 83-95, 1975.

[Newell 1975] A. Newell, "A Tutorial on Speech Understanding Systems," *Reddy*, pp. 3-54, 1975.

[Newell *et al.* 1971] A. Newell, J. Barnett, J. W. Forgie, C. Green, D. Klatt, J. C. R. Licklider, J. Munson, R. Reddy, W. Woods, *Speech Understanding Systems: Final Report of a Study Group*, Carnegie-Mellon Univ., Pittsburgh, Pa., May, 1971.

[Paxton 1974] W. H. Paxton, "A Best-First Parser," *Erman*, pp. 218-225, 1974.

[Paxton 1976a] W. H. Paxton, "A Framework for Language Understanding," Tech. Rep., Artificial Intelligence Center, Stanford Research Institute, June, 1976.

[Paxton 1976b] W. H. Paxton, "Experiments in Speech Understanding System Control," Tech. Rep., Artificial Intelligence Center, Stanford Research Institute, August, 1976.

[Paxton *et al.* 1975] W. H. Paxton, and A. E. Robinson, "System Integration and Control in a Speech Understanding System," Tech. Rep., Artificial Intelligence Center, Stanford Research Institute, September, 1975.

[*Reddy*] D. R. Reddy, ed., *Speech Recognition: Invited Papers of the IEEE Symposium*, New York, N. Y.: Academic Press, 1975.

[Reddy 1976] D. R. Reddy, "Speech Recognition by Machine: A Review," *Proc. IEEE*, Vol. 64, No. 4, pp. 501-531, April, 1976.

[Reddy *et al.* 1973a] D. R. Reddy, L. D. Erman, R. D. Fennell, and R. B. Neely, "The Hearsay Speech Understanding System: An Example of the Recognition Process," *IJCAI3*, pp. 185-193, 1973.

[Reddy *et al.* 1973b] D. R. Reddy, L. D. Erman, and R. B. Neely, "A Model and a System for Machine Recognition of Speech," *IEEE Trans. Audio and Electroaco.*, Vol AU-21, No. 3, pp. 229-238, June, 1973.

[Reddy *et al.* 1974] D. R. Reddy, and A. Newell, "Knowledge and its Representation in a Speech Understanding System," *Gregg*, pp. 253-285, 1974.

[Reddy *et al.* 1975] D. R. Reddy, and L. D. Erman, "Tutorial on System Organization for Speech Understanding," *Reddy*, pp. 457-480, 1975.

[Rieger 1975] C. H. Rieger, "Conceptual Memory and Inference," *Schank*, pp. 157-288, 1975.

[Riesbeck 1975a] C. K. Riesbeck, "Conceptual Analysis," *Schank*, pp. 83-156, 1975.

[Riesbeck 1975b] C. K. Riesbeck, "Computational Understanding," *Schank & Nash-Webber*, pp. 15-19, 1975.

[Rovner *et al.* 1974] P. Rovner, B. Nash-Webber, and W. A. Woods, "Control Concepts in a Speech Understanding System," *Erman*, pp. 267-272, 1974; also *Martin & Reddy*, pp. 136-140, 1975.

[*Rustin*] R. Rustin, ed., *Natural Language Processing*, New York, N. Y.: Algorithmics Press, 1973.

[*Schank & Colby*] R. C. Schank, and K. M. Colby, eds., *Computer Models of Thought and Language*, San Francisco, Ca.: W. H. Freeman and Company, 1973.

[*Schank*] R. C. Schank, ed., *Conceptual Information Processing*, Amsterdam: North Holland, 1975.

[*Schank & Nash-Webber*] R. C. Schank, and B. L. Nash-Webber, eds., *Theoretical Issues in Natural Language Processing*, Mathematical Social Sciences Board, Cambridge, Ma., June, 1975.

[Schank 1971] R. C. Schank, "Finding the Conceptual Content and Intention in an Utterance in Natural Language Conversation," *IJCAI2*, pp. 444-454, 1971.

[Schank 1973] R. C. Schank, "Identification of Conceptualizations Underlying Natural Language," *Schank & Colby*, pp. 187-247, 1973.

[Schank 1975a] R. C. Schank, "The Structure of Episodes in Memory," *Bobrow & Collins*, pp. 237-272, 1975.

[Schank 1975b] R. C. Schank, "MARGIE, The Conceptual Approach to Language Processing, and Conceptual Dependency Theory," *Schank*, pp. 1-82, 1975.

[Schank *et al.* 1973] R. C. Schank, N. Goldman, C. J. Rieger, and C. Riesbeck, "MARGIE: Memory, Analysis, Response Generation and Inference on English," *IJCAI3*, pp. 255-261, 1973.

[Schank *et al.* 1975] R. C. Schank, and R. P. Abelson, "Scripts, Plans, and Knowledge," *IJCAI4*, pp. 151-157, 1975.

[Siklossy *et al.* 1972] L. Siklossy, and H. A. Simon, "Some Semantic Methods for Language Processing," *Simon & Siklossy*, pp. 44-66, 1972.

[Simmons 1973] R. F. Simmons, "Semantic Networks: Their Computation and Use for Understanding English Sentences," *Schank & Colby*, pp. 63-113, 1973.

[Simmons *et al.* 1971] R. F. Simmons, and B. C. Bruce, "Some Relations Between Predicate Calculus and Semantic Net Representations of Discourse," *IJCAI2*, pp. 524-530, 1971.

[*Simon & Siklossy*] H. A. Simon, and L. Siklossy, eds., ***Representation and Meaning: Experiments with Information Processing Systems***, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1972.

[Walker 1974] D. E. Walker, "The SRI Speech Understanding System," *Erman*, pp. 32-37, 1974.

[*Waltz*] D. L. Waltz, ed., "Natural Language Interfaces," *SIGART Newsletter*, No. 61, pp. 16-65, February, 1977.

[Weizenbaum 1966] J. Weizenbaum, "ELIZA--A Computer Program for the Study of Natural Language Communication Between Man and Machine, *Comm. ACM*, Vol. 9, No. 1, pp. 36-45, January, 1966.

[Wilks 1973a] Y. Wilks, "Understanding Without Proofs," *IJCAI3*, pp. 270-277, 1973.

[Wilks 1973b] Y. Wilks, "The Stanford Machine Translation Project," *Rustin*, pp. 243-290, 1973.

[Wilks 1974] Y. Wilks, "Natural Language Understanding Systems Within the AI Paradigm," Tech. Rep., Computer Science Dept., Stanford Univ., December, 1974.

[Wilks 1975a] Y. Wilks, "Primitives and Words," *Schank & Nash-Webber*, pp. 42-45, 1975.

[Wilks 1975b] Y. Wilks, "Methodology in AI and Natural Language Understanding," *Schank & Nash-Webber*, pp. 144-147, 1975.

[Wilks 1976] Y. Wilks, "Parsing English," *Charniak & Wilks*, pp. 89-100 & 155-184, 1976.

[Winograd 1973] T. Winograd, "A Procedural Model of Language Understanding," *Schank & Colby*, pp. 152-186, 1973.

[Woods 1970] W. A. Woods, "Transition Network Grammars for Natural Language Analysis," *Comm. ACM*, Vol. 13, No. 10, pp. 591-606, October, 1970.

[Woods 1973] W. A. Woods, "An Experimental Parsing System for Transition Network Grammars, *Rustin*, pp. 111-154, 1973.

[Woods 1974] W. A. Woods, "Motivation and Overview of BBN SPEECHLIS, An Experimental Prototype for Speech Understanding Research," *Erman*, pp. 1-10, 1974; also *Martin & Reddy*, pp. 2-10, 1975.

[Woods 1975a] W. A. Woods, "What's in a Link: Foundations for Semantic Networks," *Bobrow & Collins*, pp. 35-82, 1975.

[Woods 1975b] W. A. Woods, "Syntax, Semantics, and Speech," *Reddy*, pp. 345-400, 1975.

[Woods 1977] W. A. Woods, "A Personal View of Natural Language Understanding," *Waltz*, pp. 17-20, 1977.

[Woods *et al.* 1976] W. A. Woods, M. Bates, G. Brown, B. Bruce, J. W. Klovstad, and B. Nash-Webber, "Uses of Higher Level Knowledge in a Speech Understanding System: A Progress Report," *ICASSP*, pp. 438-441, 1976.