AD/A-003 315

NATURAL COMMUNICATION WITH COMPUTERS. VOLUME I. SPEECH UNDERSTANDING RESEARCH AT BBN

William A. Woods, et al

Bolt Beranek and Newman, Incorporated

Prepared for:

Advanced Research Projects Agency

December 1974

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE

Unclassified

Security Classification

AD/A-003 315

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, MA 02138

2a. REPORT SECURITY CLASSIFICATION

unclassified

2b. GROUP

3. REPORT TITLE

NATURAL COMMUNICATION WITH COMPUTERS
Final Report - Volume I
Speech Understanding Research at BBN -- October 1970 to December 1974

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report (Technical) October 1970 - December 1974

5. AUTHOR(S) (First name, middle initial, last name)

William A. Woods, project scientist--Madeleine A. Bates, Bertram C. Bruce, John J. Colarusso, Craig C. Cook, Laura Gould, David L. Grabel, John I. Makhoul, Bonnie L. Nash-Webber, Richard M. Schwartz, Jared J. Wolf

6. REPORT DATE

December 1974

7a. TOTAL NO. OF PAGES

271

7b. NO. OF REFS

51

8a. CONTRACT OR GRANT NO.

DAHC15-71-C-0088

b. PROJECT NO.

c. order no. 1697

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

BBN Report No. 2976

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

ARPA
1400 Wilson Boulevard
Arlington, VA 22209

13. ABSTRACT

The report covers the development of the BBN speech project over the last four years from its early beginnings as part of the natural language understanding research at BBN prior to the inception of the ARPA Speech Understanding Project. At this point, the project is in the middle of the 5-year program projected by the ARPA Speech Understanding Research Steering Committee. This report is a final report on the first phase of this project and marks the transition of the Speech Project from a part of a larger contract on Natural Communications with Computers to a separate contract of its own.

A portion of the material presented here consists of adaptations of previously published papers and reports, expanded and modified to bring them up to date. There is much additional material however, which has not yet been published elsewhere. This includes many of the details of operation of the individual components and the description of the new travel budget problem domain and the pragmatics component.

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Acoustics						
Acoustic Transcription						
Artificial Intelligence						
Automatic Speech Understanding						
Case Frames						
Computational Linguistics						
Computational Semantics						
Data Structures						
Evaluating Speech Understanding Systems						
Incremental Simulation						
Lexical Retrieval						
Natural Language Processing						
Parser						
Parsing						
Phonetics						
Phonological Rules						
Semantic Networks						
Semantics						
SPEECHLIS						
Speech Recognition						
Speech Understanding						
Speech Understanding Research						
Speech Understanding Systems						
Syntax						
Transition Network Grammars						

This report is one of five volumes which compose the final report of work performed over a four year period by Bolt Beranek and Newman Inc. under contract DAHC15-71-C-ØØ88, Natural Communications with Computers. This work was supported by the Defense Advanced Research Projects Agency under ARPA order number 1697. Because of the wide spectrum of research activites performed, the final report has been structured as follows:

<u>Title</u>	<u>Volume</u>
Speech Understanding Research at BBN	I
Speech Compression at BBN	II
Distributed Computation Research at BBN	III
ARPANET TENEX	IV
INTERLISP Development and Automatic Programming	V

NATURAL COMMUNICATION WITH COMPUTERS

Final Report - Volume I

SPEECH UNDERSTANDING RESEARCH AT BBN

October 1970 to December 1974

William A. Woods
Project Scientist

M. Bates
B. Bruce
J. Colarusso
C. Cook
L. Gould
D. Grabel
J. Makhoul
B. Nash-Webber
R. Schwartz
J. Wolf

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

This research was supported by the Advanced Research Projects Agency under ARPA Order No. 1697; Contract No. DAHC15-71-C-0088.

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

TABLE OF CONTENTS

Preface

The report covers the development of the BBN speech project over the last four years from its early beginnings as part of the natural language understanding research at BBN prior to the inception of the ARPA Speech Understanding Project. At this point, the project is in the middle of the 5-year program projected by the ARPA Speech Understanding Research Steering Committee. This report is a final report on the first phase of this project and marks the transition of the Speech Project from a part of a larger contract on Natural Communications with Computers to a separate contract of its own.

A portion of the material presented here consists of adaptations of previously published papers and reports, expanded and modified to bring them up to date. There is much additional material however, which has not yet been published elsewhere. This includes many of the details of operation of the individual components and the description of the new travel budget problem domain and the pragmatics component.

Acknowledgment

The gratitude of the entire speech group goes to Mrs. Linda Amsden for her exceptional diligence, competance, and good humor shown throughout the assembly of this report. She has been our reference librarian, copy editor, teco wizard and secretary all rolled into one.

I. INTRODUCTION AND OVERVIEW

A. Overview of the Project

1. Context

The BBN Speech Understanding Project is currently in the middle of a 5-year program to develop a continuous speech understanding system. The BBN effort is part of the ARPA Speech Understanding Research (SUR) project supervised by the ARPA Speech Understanding Research Steering Committee, which encompasses the work of five major "systems builders": BBN, Systems Development Corporation, Stanford Research Institute, Carnegie-Mellon University, and (formerly) Lincoln Laboratory. The project also includes various specialist contractors, including Haskins Laboratories, Speech Communications Research Laboratory, UNIVAC, and the University of California at Berkeley.

According to the guidelines for the project set down by the Steering Committee, during the first two years of the project, each of the systems builders was to construct a complete, but preliminary speech understanding system. This would demonstrate their competence and readiness for the second half of the project and permit the final speech understanding systems to benefit from their first round of mistakes. In November of 1973, the five systems were evaluated by the Steering Committee and recommendations were made to ARPA for the structure of the continuation of the project. As a result of this evaluation, BBN, a project at Carnegie-Mellon University, and a combined project between SRI and SDC, were selected to continue the development of total speech understanding systems.

2. Emphasis

The thrust of the BBN speech understanding project has been towards two goals. First, we are attempting to use as much specialized and sophisticated knowledge as possible during the acoustic/phonetic analysis of the speech signal in order to obtain the maximum information from the acoustic signal. Second, we are attempting to discover effective techniques for using higher level linguistic information such as knowledge of vocabulary, syntax, semantics, and pragmatics in order to compensate for ambiguity and indeterminacies in the acoustic/phonetic analysis. Our project differs from other speech understanding projects in the level of sophistication which we are attempting to apply to the acoustic/phonetic analysis problem and in the syntactic fluency and semantic range that we are aiming for in our higher level linguistic components.

As the size of the vocabulary, the fluency of the syntax, and the scope of the semantics increase, they become less constraining, and the importance of obtaining high quality acoustic/phonetic analyses increases. The BBN speech understanding project is oriented toward finding the limits of our abilities to use sophisticated acoustic/phonetic processing and higher level linguistic constraints to handle difficult problems, and toward discovering techniques for dealing with such problems.

While the mandate of the current speech project permits the use of very tightly constrained syntax and semantics to compensate for uncertainties in acoustic/phonetic decoding, the narrow use of such constraints will also limit the possible applications for speech understanding systems. Therefore we have been concerned with the long range objectives of determining required techniques for dealing with the cases where the syntax becomes more fluent and the semantics less limited. We have been aiming for a system that can understand natural English with fairly broad fluency, with a fairly powerful range and complexity of semantic concepts. Our major interest, and I believe the principal product of the current ARPA SUR project, is to gain an understanding of the tradeoffs in performance as a function of vocabulary size, syntactic fluency, semantic range, and quality of acoustic/phonetic performance. Consequently, we have taken seriously the deemphasis on immediate real-time requirements, given by Dr. Lawrence Roberts in his initial charge to the Speech Understanding Study Group in Pittsburgh [33]. We are shooting for algorithms which are capable of being implemented in near real time on machines with speeds that are expected to exist in the near future, but not limiting ourselves to techniques which can be done in real time on present machines. Since we are constructing a system as a breadboard for experimenting with sophisticated techniques, our primary concerns in this 5-year program are with designing algorithms which are capable of being run in close to real time when carefully implemented on appropriate hardware and with attaining sufficient speed in our breadboard system to perform desired experiments.

To summarize then, the emphasis of the BBN project is to discover what is necessary to do the difficult jobs rather than determining the power of limited mechanisms.

3. Synopsis of Research to Date

BBN's effort in continuous speech understanding began with a set of spectrogram reading experiments by Klatt and Stevens at M.I.T. [21]. These experiments consisted of two phases. During the first phase, each experimenter attempted to perform an objective phonetic transcription of the utterance without attempting to guess the content of the utterance or the words involved. This objectivity was enhanced by looking at the spectrogram through a narrow slot which uncovered only a few hundred milliseconds of signal at a time (about the amount for three successive phonemes). An experimenter was permitted some vagueness in his transcription, depending on his ability to identify unambiguously the phoneme under consideration. For example, he could merely describe a given segment as a back vowel, or as a voiced plosive, if the acoustic cues in the signal did not give him sufficient confidence to be more precise. He was also allowed some vagueness in postulating the existence of a segment by indicating it as optional. That is, if he were uncertain whether a given portion of the signal was a separate phoneme, part of an adjacent one, or a transitional segment, he could both describe the segment as if it were a distinct phoneme and also indicate its possible non-existence.

During the second phase of these experiments, the researchers were able to employ higher-level linguistic constraints in producing their transcriptions. Using a computerized retrieval system written at BBN to access the lexicon on the basis of partial phonetic information similar to that used in their first-phase efforts, they attempted to transcribe the utterance into a string of English words. During this second phase, they were free to use all of their intuition about English syntax and semantics in attempting to reconstruct the sentence. The results of the experiment indicated that while their error rate was 25-30% in the objective phonetic transcription phase (even with the latitude permitted by partial or optional segment specifications), in the second phase they were able to identify the words of the utterance with a 96% success rate. This experiment tended to verify our assumption that knowledge from the higher level linguistic components can compensate for acoustic indeterminacies in the acoustic/phonetic transcription.

A side benefit of the Klatt and Stevens experiments were the computer protocols of their second-phase sessions. Retrospective analysis of these protocols provided valuable insights into techniques used by these human spectrogram readers in attempting to assign interpretations to speech utterances. For example, we could see places where the experimenter abandoned a given portion of the utterance and skipped to the right to analyze a different portion, returning later to the troublesome portion, bolstered by additional information about the utterance. We also noticed that the experimenters never consulted the lexical retrieval programs for small function words, but rather (presumably) merely recognized them in the appropriate places. These and other observations about their strategies were sufficiently suggestive to enable us to formulate a general overview of a speech understanding system. However, the information present in those protocols left many questions unanswered.

4. Incremental Simulation

In order to go further along the lines suggested in the initial Klatt and Stevens experiments, we decided to begin the design/construction of the BBN speech system by means of an approach which we dubbed "incremental simulation". It consists of "implementing" the various components of the eventual overall speech understanding system with combinations of human simulation and computer programs. The human simulator for a given component is simultaneously concerned with a number of tasks:

(a) effectively performing the role of his component in understanding the utterance,

(b) gaining insight into the problems that his component is required to solve,

(c) trying to devise algorithmic procedures to enable a computer program to effectively perform this role, and

(d) trying out these mechanical algorithms by hand and evaluating their effectiveness.

As portions of the strategy associated with his component become well understood and mechanical, he constructs computer programs to carry out those functions, and gradually builds himself out of the component, remaining only in a role of monitoring performance and considering techniques for improving performance. This mode of system development permits the system designer to gain immediate insight into the problems that he needs to deal with and to discover shortcomings in proposed solutions without a lengthy period for design and implementation of the hypothesized "solution". In the course of a single simulation, the designer/simulator for a given component can formulate and discard several possible techniques for dealing with the problems.

Our first attempt at this mode of system design consisted of several steps:

(a) constructing a cr'de mechanical word matching algorithm to supplement the lexical retrieval algorithm already implemented,

(b) "implementing" an acoustical feature extraction component by simulating it with a human spectrogram reader connected to the system by a teletype l..nk,

(c) constructing a bookkeeping component to keep track of what had been done, and

(d) simulating the syntactic, semantic, pragmatic and control components with a single experimenter.

It was our goal to develop a feeling for the general overall control strategies which are effective in understanding an utterance, given the types of acoustic/phonetic segmentation information provided by the simulated acoustic feature extraction component. These simulations gave us a good understanding of the problems of continuous speech understanding for fairly fluent syntax and moderately constrained semantics. A paper presented at the Third International Joint Conference on Artificial Intelligence and subsequently published in the journal Artificial Intelligence [50] describes and illustrates this technique. Subsequent simulations involving separate individuals for the control, syntax, and semantics components developed the basic structure for the current BBN Speech Understanding System. Details of these structures are still under evaluation, however, and they change as we gain further experience running the system and as the capabilities of the individual components grow.

5. Signal Processing

Concurrently with the incremental simulation experiments used to develop insights into the organization of the control component and the various higher level linguistic components, a sophisticated display-oriented signal processing facility was constructed using an IMLAC PDS-1 display processing computer connected remotely to the BBN PDP-10. [See Appendix A for further discussion of the hardware for this facility.] This system has been used to develop a number of new techniques in digital signal processing (based on linear prediction) for speech understanding and to search for useful parameters which could be computed from the speech signal and used as cues to the identity of speech sounds. Results of this research have been published in a variety of technical reports and articles [24,25,26,27,28,29], and research using this system is continuing.

B. The Two-Year Demonstration System

Because of the necessity for demonstrating a total speech understanding system at the end of the first two years of the ARPA SUR project and also in order to gain some input data on which to test the operation of the control strategy and support from the higher level linguistic components, we accelerated our work on acoustic/phonetic segmentation and labeling to produce a temporary first-cut phonetic segmenter and labeler. The assignments of this initial segmenter and labeler were based largely on manner of articulation (stop, fricative, nasal. vowel distinctions), with place-of-articulation information for vowels, glides, and strident fricatives. This component, plus a general purpose inverse phonological rule component served as the input for the control and higher level components of the November 1973 system, demonstrated to the evaluation team of the ARPA Speech Steering Committee. A fairly detailed description of this system was presented in a collection of papers presented to the IEEE Symposium on Speech Recognition at Carnegie-Mellon University in April of 1974, many of which have been submitted for publication elsewhere. These papers have been collected together in a technical report [48] and they provide a basis of much of the current report.

We learned a number of things from the construction of this interim system. One of the notable results was the difference in segmentation errors between the automatic segmenter and labeler and the manual simulations by human spectrogram readers. Whereas the human spectrogram readers made a good number of missing segment errors, they rarely postulated extra segments. The automatic segmenter and labeler, on the other hand, made a large number of extra segment errors. In general, while humans were very good at deciding that a given phenomenon was a transitional segment or a glitch in the signal, the computerized version lacked this type of knowledge. Many cases of over-segmentation were caused by differences in onset time for the various features of a segment. For example, a [z] following an unvoiced segment may commence with unvoiced frication with the voicing beginning 10-20 milliseconds later. To an uninitiated segmenter and labeler this looks like an [s] followed by a [z]. This and other phenomena were identified, and some inverse phonological rules were devised to correct for the effects (e.g. an inverse rule that optionally transforms [s z] into [z]). Because the correct place for such knowledge to reside seems to be in the acoustic/phonetic decoding routines themselves, we plan to move it there in our new acoustic/phonetic analyzer, leaving the phonological rules to account for genuinely rule-driven phonological phenomena.

We have also learned some things about the operation of the higher level components from experimenting with the November system, both on automatically and manually produced segment lattices. We have identified a number of cases where either prosodic or pragmatic information is required to reject erroneous interpretations that satisfy all forseeable syntactic and semantic conditions, and we have identified some general pragmatic principles which would account for these cases. We also have speculated on possible prosodic cues which could resolve these cases, and we have cooperated with Medress and Lea at UNIVAC in having these sentences analyzed by their prosodic analysis routines.

Moreover, we are using experiments with this system to continue to specify and refine our control strategy. A current problem that we would like to solve is whether we can use information from a rejected theory about the utterance to suggest better ones, rather than simply abandoning it to search for better theories. We have encountered a number of cases where the first total theory developed was correct except for one or two words. We would like to identify and use the correct parts of such a theory to deduce a correct total theory, so as to reduce the time required by our current technique.

C. Beyond the Two-Year System

Since the November demonstration, work on the project has concentrated on the design of the system which is to be demonstrated at the end of the fifth year. This includes the redesign and construction of both a new segmentation and labeling component and a new lexical retrieval and word matching component, the design and implementation of a second domain of discourse, and the development of a number of experimental features such as a sophisticated, analysis-by-synthesis word verification component. Work will continue as it has been in attempting to develop effective control strategies for integrating the knowledge from the various higher level linguistic components and for structuring those components for maximum efficiency, and we are beginning to design a more systematic pragmatics component.

In subsequent sections, we will present in more detail a description of the November system and what we have learned from it, a discussion of our recent work, and projections about the future system.

D. Publications

To date the project has resulted in a number of technical reports, published articles, and chapters for books. These include a definitive volume on linear predictive analysis by Makhoul and Wolf [25], an introductory article on inference problems in speech understanding by Woods and Makhoul [50], tutorial papers by Makhoul and by Woods in Raj Reddy's book on Speech Understanding [49], and a chapter by Nash-Webber on semantics and speech understanding in Representation and Understanding by Bobrow and Collins (in press) [52]. We give in Appendix B a complete list of the publications resulting from the project to date.

E. Motivation and Overview of the November 1973 System

1. Introduction

This section describes the November 1973 version of our computer system for carrying out research in continuous speech understanding. The system is a research prototype of an intelligent speech understanding system which makes use of advanced techniques of artificial intelligence, natural language processing, and acoustical and phonological analysis and signal processing in an integrated way to determine an interpretation of a continuous speech utterance which is both syntactically and semantically plausible and consistent with the acoustic-phonetic analysis of the input signal.

We take as a point of departure that the information required to produce the correct interpretation of an utterance is not completely and unambiguously encoded into the speech signal, but rather that knowledge of the vocabulary and of syntactic, semantic, and pragmatic constraints of the language are used to compensate for uncertainties and errors in the acoustic realization of the utterance. This fact seems appropriately substantiated by human perceptual performance [42] and by Klatt and Stevens's spectrogram reading experiments [21]. In the latter, human experts attempting to decipher spectrograms achieved error rates of approximately 25% in "partial" phonetic transcription based on spectrographic evidence alone but were 96% successful in identifying the words of the utterances when permitted to make use of knowledge of the vocabulary and of syntactic and semantic constraints. It is the matching of human performance in these experiments towards which the BBN speech understanding system (dubbed SPEECHLIS) aspires.

In a previous paper [50] we described the method of "incremental simulation" which we have used to get a feeling for the types of interaction among the different sources of knowledge used during the understanding of a speech signal. In that article, we postulated the decomposition of a speech understanding system into separate components and presented an illustrative example of their interaction in the analysis of an utterance. We also discussed the types of inference capabilities which would be required from the different components in a mechanical speech understanding system. In this paper we will describe how we have attempted to embody those capabilities in SPEECHLIS.

Whereas this chapter gives an overview of the system and its motivations, subsequent sections will give more detailed descriptions of the operations of individual components.

2. Domain of Discourse

If one is to use knowledge of vocabulary, syntax, and semantics in a speech understanding system, it is necessary to select what vocabulary, syntax, and semantics to deal with. For our initial domain, because of its ready availability and its sophisticated syntax and semantics, we selected the domain of the LUNAR system [46,51], a natural English question-answering system dealing with chemical analyses of the Apollo 11 moon rocks. The LUNAR system understands and answers such questions as:

> "What is the average concentration of rubidium in high-alkali rocks?"
>
> "List potassium/rubidium ratios for samples not containing silicon."
>
> "how many rocks contain greater than 15% plagioclase?"

It contains a vocabulary of approximately 3500 words and a grammar for an extensive subset of general English. For our initial speech system, we selected a subset of approximately 250 words from LUNAR's vocabulary and a subgrammar of more restricted English from its grammar. In the future we intend to increase our vocabulary to over 1000 words, extend our grammar to include the entire LUNAR grammar, and include several additional domains of discourse unrelated to lunar geology. We have already begun the inclusion of a travel budget management domain.

3. Knowledge Gathering

In order to gain an understanding of the types of interaction required in using higher level linguistic knowledge to augment the (acoustic) analysis of the speech signal, we ran "incremental simulations" of the speech understanding system by "implementing" its components as combinations of computer programs and human simulators. From these simulations, the following general conclusions were reached:

(a) Small function words such as "a", "of", "the", etc., which are generally unstressed and short, have a high probability of matching accidentally in the signal. They are therefore unreliable cues by themselves on which to make a decision about an utterance and are unprofitable to look for on a "bottom up" or analytical scan of the utterance. However, when the hypothesized content words of the utterance are being parsed according to a grammar of English, syntactic knowledge is able to predict those places where such function words might occur, and in many cases, further semantic information is capable of predicting which function words are likely.

(b) It is not generally possible with the current estimated level of performance of the acoustic analyzer to distinguish correct from incorrect word matches by acoustic word match scores alone. When a threshold of acoustic match quality is set sufficiently low to accept a high proportion of the correct word matches, a large number of accidental matches of other words are also accepted. The ratio of extraneous matches to correct ones depends on the setting of the threshold (as the threshold is relaxed the ratio gets higher), but for reasonable settings it may be on the order of 20 to 1. Moreover, it appears to be impossible to set the threshold sufficiently low to guarantee acceptance of all correct word matches without swamping the system with extraneous accidental matches. However in human simulations, although it required considerable thrashing around in difficult cases, it was generally possible to go back to selected regions of the utterance after partial lexical, semantic, and syntactic analysis and perform additional phonological and phonetic analysis and/or word matching to obtain the correct words. Although we are attempting to provide such processes in our system, they are likely to be more combinatoric in their searching for possibilities than the human simulation. It is far too early to predict the success of their performance.

(c) The process of inferring an interpretation from a speech signal is inherently non-deterministic. That is, it is frequently not possible to make a particular decision (such as which of several matching words is the correct one at a given position) without making an assumption and following out its consequences for the rest of the interpretation. Mechanisms must be provided for following out all of the alternative choices in order to find the correct interpretation.

(d) No adequate a priori order can be established for scanning the utterance (such as left-to-right) for word matches or for syntactic and semantic processing. This is because any given word may be garbled in its pronunciation or phonetic analysis, and we would like to use the successful analysis of the rest of the utterance to recover the garbled word. Hence classical left-to-right parsers will not suffice, nor will semantic interpretation rules such as those in LUNAR which are indexed solely under the head of the construction being interpreted. The head of the construction may be the word that is garbled and we may need to find the successful match of the rest of the rule in order to infer the garbled word.

(e) The space of possible alternative computation paths which could lead to an interpretation of a signal is too vast to be searched in its entirety. In fact, even the set of strategies which could be tried to get an interpretation when one has not yet been found is open-ended. Examples of these strategies include relaxing the threshold of acceptability for word matches in the utterance (or in portions of it), trying the next best acoustical analysis of a given segment or combination of them, looking for possible alternative ways to segment the utterance into phoneme sequences, deciding to accept an interpretation of the utterance even though it is not syntactically well-formed, or deciding to accept an interpretation which is not semantically meaningful. (I heard what you said but it doesn't make sense.) Because of the openendedness of this search space, it is essential to devise strategies for searching it which devote their effort to the regions of the space most likely to yield the best interpretation and work out from these toward less and less likely interpretations. This requires the use of decision criteria to evaluate the goodness of a word match, and to weigh the alternatives of, say, a more grammatical interpretation with poorer word matches against a sequence of better word matches which doesn't parse or doesn't make sense. It is critical to know the difference between reliable and unreliable clues and to juggle competing alternative partial interpretations so as to continually devote effort to the best ones.

(f) Even with strategies for selectively pursuing alternatives according to their likelihood of success, the combinatorics of the situation are such that the system will be swamped with alternative possibilities unless special techniques are used to keep potentially different alternatives merged for processing operations for which they behave identically, splitting them up only when an operation being executed has a different effect for the different alternatives. One must avoid prematurely multiplying combinations of cases. For example, one cannot afford to multiply out all of the possible sequences of phonemes which could cover the utterance.

The system which we have been developing has been designed to meet these requirements.

F. Components of the System

1. Principal Knowledge Components

As a consequence of examining the protocols and results of the Klatt and Stevens experiments it was apparent that their performance was based on the capabilities of at least six conceptually distinguishable components

(a) an acoustic feature extraction component which performs the equivalent of a first-pass segmentation and labeling of the acoustic signal into partial phonetic descriptions, probably taking into account knowledge of phonological rules.

(b) a lexical retrieval component which, on the basis of knowledge of the vocabulary and partial phonetic descriptions, retrieves words from the lexicon to be matched against the input signal.

(c) a word verification component which, given a particular word and a particular location in the input signal, determines the degree to which the word matches the signal.

(d) a syntactic component which is capable of judging grammaticality of an hypothesized interpretation of the signal and of proposing words or syntactic categories to extend a partial interpretation.

(e) a semantic component which is capable of noticing coincidences between semantically related words which have been found at different places in the signal, judging the meaningfulness of an hypothesized interpretation, and predicting particular words or specific classes of words for extending a partial interpretation.

(f) a pragmatic component, which is capable of making judgments and predictions as to the pragmatic likelihood of a given sentence being uttered by the speaker, taking into account whatever is known about the speaker and the situation.

In addition to these 6 components which correspond to some extent to different sources of knowledge that go into the determination of the preferred interpretation, there is clearly an additional component of a different sort -- namely the decision process itself. In this component, which we have called the control component, reside the strategies for infering an interpretation of the utterance, dealing with questions such as:

Where should one look for word matches first?

How much partial phonetic information is given as input to the lexical retrieval routine?

How good a word match score is required for the word to be given further consideration?

How and at what points does one use syntactic and semantic information to influence the interpretation?

How are alternative possible interpretations formed, managed, and resolved?

When should one temporarily abandon a given region of the utterance to concentrate on another region?

What information might be found elsewhere that might help, and how can it be used?

These and myriad other questions have answers (not necessarily optimal) embedded in the procedures used by the human experts to interpret the spectrograms in the Klatt and Stevens experiments. We need to capture similar strategies in the control component of our speech understanding system.

2. The Control Component

Clearly the strategies embedded in the control component, critical to the success of the system, are far from obvious. We have attempted to arrive at a reasonable set of such strategies by drawing on intuitions developed in incremental simulations. These strategies are being continually refined and extended as we gain more experience with the evolving SPEECHLIS.

The function of the control component centers around the creation, refinement, and evaluation of formal data objects called "theories", which represent alternative hypotheses about the utterance being interpreted. A theory contains the words hypothesized to be in the utterance and where they match, semantic hypotheses about how those words relate to each other, hypotheses about syntactic structure, and various scores reflecting the "likelihood" of the theory from different points of view (lexical match quality, semantic completeness, syntactic correctness, etc.). These theories generally represent only partial hypotheses, beginning with single word theories with little or no syntactic or semantic detail, constructing larger theories by refinement, and eventually building up to complete theories representing hypotheses for a sequence of words covering the entire utterance with complete syntactic structure and semantic interpretation. The task of the control component is to manage the creation and refinement of these theories, devoting its resources to expanding those theories which look best according to their various scores until one or more complete theories with acceptable scores are found. Control passes partial theories at various times to the syntactic and semantic components, which return them with evaluation scores or suspend them, after creating monitors for events (which could cause the refinement of a theory) and making proposals for word matches (which Control should recall the word matcher to look for). Monitors behave as active "demons" to give notices to Control whenever events of the type which they are looking for occur. Each monitor remembers the theory which set it and a procedure which is to be executed to assimilate the event that triggers the monitor. The result of executing this procedure will be a new refined theory which may itself set additional monitors and make proposals.

In the next few sections, we will describe in a little more detail the various components of the November 1973 system. More detailed descriptions of the individual components will be given in later chapters.

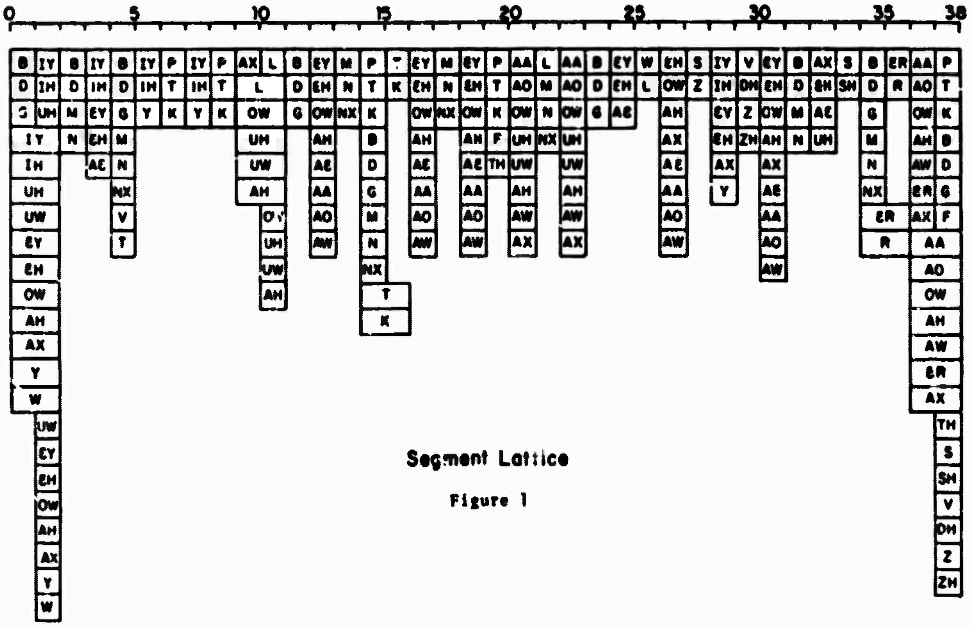
3. Acoustic-Phonetic and Phonological Analysis

In the acoustic end of our system, the speech signal is sampled at 20 kHz and stored on a disc file. All subsequent analysis is performed on the digitized signal. Using our recently developed method of "selective linear prediction" [24,25] we perform a linear predictive (LP) analysis on the 0-5 kHz region of the spectrum. Presently, almost all our parameters are based on that portion of the spectrum, the exception being a parameter giving the spectral energy between 5-10 kHz, which is used for detection of frication. The parameters used in our segmentation and feature extraction are based on: energy of the signal, energy of the differenced signal, low-frequency energy, the first autocorrelation coefficient, the normalized LP error, energy-sensitive and energy-insensitive spectral derivatives, fundamental frequency, frequencies of a two-pole LP model [26] and poles of a 14-pole LP model. We have developed an initial set of algorithms for the nondeterministic segmentation of the utterance into a segment lattice. Associated with each segment boundary are confidence measures that reflect the likelihoods of that point in the utterance being a segment boundary and of it being a word boundary. Another set of algorithms performs a feature analysis on each of the segments. We have concentrated thus far on the recognition of manner of articulation, e.g. vowel, nasal, lateral. retroflexed, plosive, fricative, voiced/unvoiced. The only place of articulation recognition that we do is performed on the vowels and strident fricatives. Confidence estimates for each of the features and for the entire segment are also given.

The output of the acoustic-phonetic analysis is in the form of a segment lattice, an example of which is illustrated in Figure 1. It compactly represents all of the possible alternative segmentations of the utterance and the alternative identities of the individual segments. This lattice is processed by a phonological rule component which augments the lattice with branches for possible underlying sequences of phonemes which could have resulted in the observed acoustic sequences. We associate with each added branch a predicate function which is later used by the word matcher to check for the applicability of the given phonological rule based on the specific word spelling and the necessary context. In this manner, the phonological rules are both analytic and partially generative. Other generative rules can be applied ahead of time to the dictionary phonemic spellings of words -- such rules have been done manually in our November 1973 system.

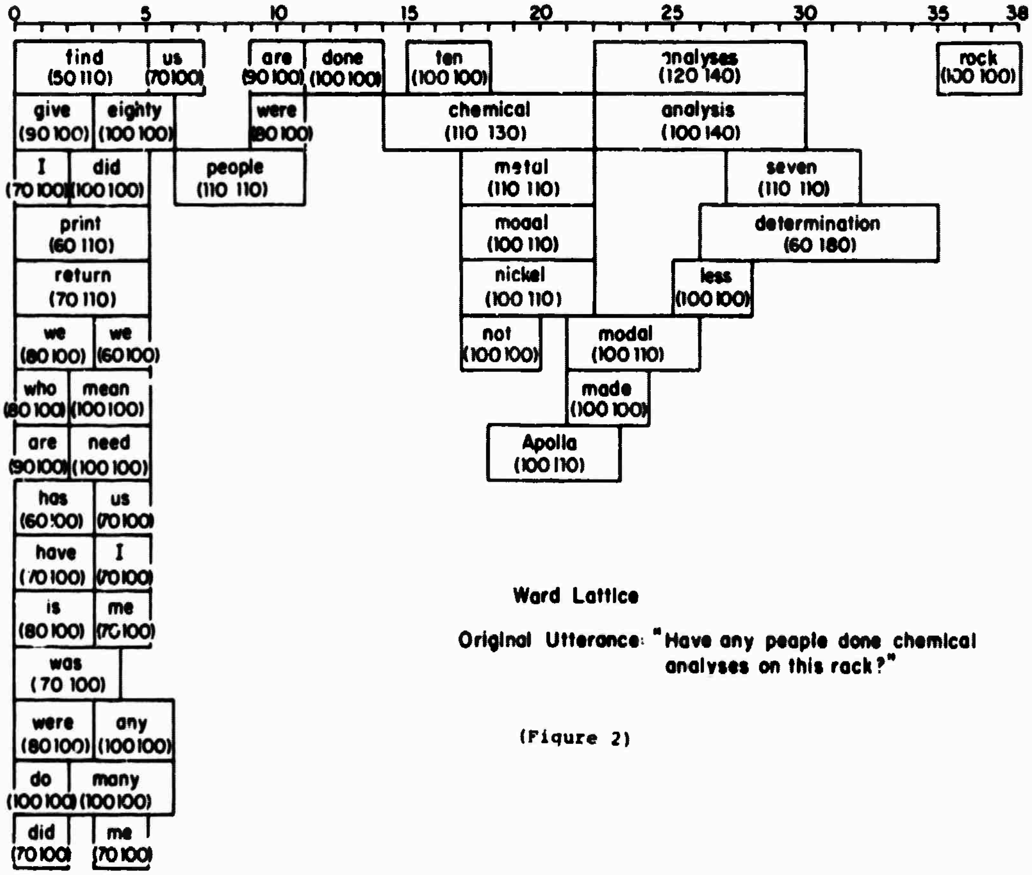
4. Higher Level Linguistic Constraints

The current lexical retrieval and word matching component makes use of a phonetic similarity matrix for evaluating non-exact phoneme matches, phonologically motivated deletion likelihoods for each of the phonemes in a word, and rudimentary duration cues based on stress marks in the phonemic spelling of the word. Words with three or more phonemes which score above a threshold of match quality are placed in a "word lattice," an example of which is illustrated in Figure 2. They are given individually to the semantic component which constructs a one-word theory for each content word, monitors for words that could be semantically related to the given one, and generates events for each detected coincidence between two or more semantically related words or concepts.



Segment Lattice

Figure 1



Word Lattice

Original Utterance: "Have any people done chemical analyses on this rock?"

(Figure 2)

Each word is also checked for matching inflectional endings, and verbs are checked for possible auxiliaries to their left and at the beginning of the utterance.

The semantic coincidence events are sorted by the control component in order of their likelihood scores and at appropriate times are returned to Semantics for the construction of larger theories. In this way, multiple word theories are constructed which consist of semantically related content words which match well acoustically. When a theory becomes maximal (i.e., Semantics has no further words to add to it), it is passed to Syntax for syntactic evaluation. In addition to evaluation, Syntax picks up further words from the word lattice and proposes words (especially function words) to fill the gaps between the words originally provided in the theory. Syntax also monitors for syntactic categories of words which it could use to fill gaps. When Syntax completes a constituent (such as a noun phrase) it calls Semantics directly to verify the consistency between the syntactic structure of the constituent and the semantic hypotheses for its words.

The control strategy maintains a list of active theories, pending events, and proposed words and classes -- all ordered by estimates of likelihood -- and determines which theory/event/proposal to work on next at each point.

Some pragmatic inferences have been identified and embedded in the control strategy, but no systematic pragmatics component has been incorporated. The construction of semantic procedures for answering questions using the data base has not yet been implemented, since we have previously done this once with the LUNAR system and have been devoting our effort instead to the new aspects of the system.

5. Preliminary Results Obtained

Since the current phase of the BBN speech project is more concerned with finding the problem areas and developing possible solution techniques, it is premature to expect statistical results such as percentage of utterances successfully understood. Rather, the principal product of the research at this point consists of experiences that suggest experiments yet to be done and techniques whose effectiveness has yet to be fully measured. The following are some examples:

(a) The inclusion in the word matching function of simple duration checks for stressed phonemes and of deletion probabilities for each phoneme decreased the scores of many of the accidental word matches without effectively lowering the scores of the correct word matches. This suggests a host of experiments -- how much improvement can you obtain? -- with what cost?

(b) The ambiguities of segmentation and labeling of the acoustic signal can result in the same word matching the input signal in approximately the same place in several different ways with slightly different end points and slightly different scores. From the point of view of the semantic associations invoked, these word matches are all the same and should not be dealt with by separate theories, one for each such match. This has resulted in the creation of a "fuzzy word match" which lumps together equivalent word matches into a single entity which is dealt with by Semantics as a single word match with ambiguous end points. This greatly reduces the number of theories processed.

(c) A similar phenomenon occurs when several words from a single semantic class all match the signal at the same point (for example the pronouns "I", "we", and "us"). Again, since Semantics will initially do the same thing for each such word, these are grouped together into a "clump" which is treated as a single word until such time as later processing splits it up.

(d) Certain acoustic-phonetic facts which are not currently dealt with by the segmenting and labeling component can cause recognizable pathologies at later stages of processing. For example, the fact that voicing frequently drops out before the end of frication in a voiced fricative followed by an unvoiced segment may cause the segmenter to recognize a segment sequence [z][k] as a sequence [z][s][k] causing word matches for "samples" and "contain" which should be adjacent to have a spurious [s] segment between them. This problem could be dealt with either by improving the initial segmentation and labeling algorithm, or by an analytic phonological rule to combine the voiced and unvoiced fricative in this context into a single voiced fricative, or by a higher level word adjacency test which considers two words to be adjacent if a spurious segment between them can be accounted for as an expected transition segment. This suggests experiments to be peformed when the system is more fully developed to determine the most effective place to deal with this and similar problems.

(e) It is possible to get alternative interpretations with almost equally good lexical, syntactic, and semantic evaluations -- even two interpretations with exactly the opposite meaning. In all such situations which we have witnessed, there has been other information (such as prosodic or pragmatic information) available to make a choice, but it seems clear that the information which could be so used is open ended, and it is not clear how much is required in order to get acceptable performance even for a 250 word vocabulary, much less a 1000 word vocabulary.

The list of such questions which are being raised could go on and on. However, the above list should be suggestive of the types of results which we are obtaining.

G. A Sample of Current Performance

1. Issues of Evaluation

We have outlined the methodology and the current state of a project to develop an advanced speech understanding system via continual incremental improvements to initially crude components. An important consideration for such a program is a method for evaluating the progress of this evolutionary development in terms of the performance of the system or of its parts. How does one measure the improvement (or degradation) in system performance caused by a particular change to a strategy in one of the components? Although our current system has not yet reached the stage where we are prepared to run many utterances through it to compute statistics of performance, we have given some thought to what statistics of performance one would like to see and have made some initial measurements of them on test sentences.

Evaluation parameters fall into two classes, measures of precision and measures of accuracy. For example, in evaluating the performance of the segment labeler, precision measures the degree to which the label assigned uniquely specifies the phonemic identity of the segment, while accuracy measures the frequency with which the description is correct. There is clearly a tradeoff between these two measurements since one can achieve perfect accuracy by relaxing precision to the point where the description assigned is sufficiently vague to include all of the phonemes. On the other hand, one could only achieve perfect precision by choosing at every point the single most likely phoneme with a subsequent loss of accuracy. There are similar measures of precision and accuracy for the process of segmentation itself (as opposed to labeling) and the process of lexical retrieval and matching.

As a measure of precision in segmentation, we may take the branching ratio of the segment lattice, i.e. the number of segments per boundary. Accuracy in segmentation falls into two categories -- the number of missing boundaries (i.e. segment boundaries which were not identified as potential boundaries in the lattice) and the number of extra boundaries (i.e. points in the utterance identified as boundaries in the lattice which were not segment boundaries and for which there is no "bridging" segment crossing that region of the utterance).

Specific precision and accuracy measures for segment labeling are the average number of phonemes per label (i.e. the number of phonemes subsumed under the description assigned to a segment) and the average percentage of errors in labeling (when the correct phoneme is not subsumed in the assigned description).

At the lexical level, we can measure the success of the initial lexical retrieval pass in terms of the number of correct words found (out of the total number of correct words to be found -- an accuracy measure) and the "stray word ratio" (the ratio of the total number of words found to the number of correct words found -- a precision measure).

Clearly there are precision/accuracy tradeoffs throughout the system. By merely adjusting the threshold of acceptable word match quality, the number of correct words found and the stray word ratio can be altered without any change at all in the algorithm being used for word matching.

While we have not performed the necessary experiments to be able to give any conclusions about the behavior of these parameters as a function of differences in strategies, threshold levels, etc., and while the current components give only crude approximations to the performance which we expect, we have conducted a few tests which may serve as benchmarks Figure 3 gives the results of some tests (made in October, 1973) on two utterances using three different acoustic analysis methods to produce the segment lattices. The first case (manual) is the result of a human spectrogram reading as in the first phase of the Klatt and Stevens experiments. The second case (auto1) is the result of our first crude segmenting and labeling program which estimates only the manner of articulation of the segments and does not measure place of articulation. The third case (auto2) makes use of a slightly improved version (but still crude) of the segmenting and labeling program, which tracks formants and estimates place of articulation for vowels. At the bottom of Figure 3 is shown the word match score assigned by the lexical retrieval component to each of the correct words that it found. We did not run it on the auto2 lattice for utterance DWD-29.

EXAMPLE OF PERFORMANCE OF ACOUSTIC-PHONETIC PROCESSING
AND
LEXICAL RETRIEVAL SCAN FOR "GOOD" "BIG" WORDS

	DWD-18				DWD-29			
	IDEAL	MANUAL	AUTO1	AUTO2	IDEAL	MANUAL	AUTO1	AUTO2
# segs in ideal segmentation	34				27			
# missing bdries	0	0	0	0	0	0	4	4
# extra bdries	0	0	0	0	0	0	0	0
# segs/bdry	1	1.2	1.3	1.3	1	2.0	1.5	1.5
% errors	0	12%	22%	10%	0	4%	43%	30%
# phonemes/label	1	6	4	3	1	4	4	3
# words ideal	9				8			
# words ≥ 3	8				5			
# correct words found		6	5	5		5	0	
# words found total		127	130	92		238	48	
# words missed		2	3	3		0	5	
stray word ratio (# words matched/ # correct)		21	26	18		48	-	

	have	any	people	done	chemical	analyses	on	this	rock	give	me	all	lunar	samples	with	magnetite
MANUAL		100	110	100	110	120			100	90			100	120	100	140
AUTO1		90		90	110	120			100							
AUTO2		100		90	120	140			100							

Figure 3

Our current front-end analysis component tends to be better at some kinds of phonetic events than at others. This is a result of the almost encyclopedic amount of acoustic-phonetic and phonological knowledge which is required to deal with the different phenomena which can occur and the relatively short amount of time which we have had to embody this knowledge in computer algorithms. This difference is illustrated by the differences in performance between the two utterances DWD-18 ("Have any people done chemical analyses on this rock?") and DWD-29 ("Give me all lunar samples with magnetite."). The former seems to contain only phenomena with which the current programs deal reasonably well, while the latter contains such troublesome configurations as the "all lunar" sequence. In DWD-18, the performance of the auto2 acoustic analyzer is superior to that of the manual analysis in terms of the precision and accuracy measures, but its errors are slightly different from those of the manual analysis, and in particular, its resulting transcription is such that the "people" word match which was found on the manual analysis was missed for auto1 and auto2. This is due to the effect of a phonological rule which the human apparently took into account in his analysis but which the mechanical analysis component did not know about. The phonological rule component which has been implemented since these experiments were run is capable of recovering this match.

2. Performance of Syntax and Semantics

For the higher level components of Syntax and Semantics, the same types of precision and accuracy measurements no longer seem appropriate until one has processed large numbers of utterances and recorded the success rate; and even then, there is no natural notion of a precision measure. Questions of interest in the syntactic and semantic areas of the system include: how much effort is devoted to searching blind alleys before a correct interpretation of the utterance is found?, how many false interpretations are accepted in addition to (or before) the correct one?, is the correct one found at all?, etc.

While we do not begin to have, again, answers to these questions, we have run test cases which can serve as benchmarks. We will illustrate with a brief summary of the syntactic and semantic processing of a sentence DWD-24 ("How many samples contain silicon?") from a segment lattice obtained by mechanical segmentation and labeling. (Two editing changes were made to the lattice to manually simulate the effects of phonological rules.)

In the initial lexical retrieval scan of the segment lattice for this sentence, word matches for "sample", "contain", and "silicon" were found with acceptable acoustic scores, together with a number of other accidental word matches such as "contain" (in another place in the input), "occur", "occurring", "with", "content", "contents", and many others. In the formation of one-word theories, four different matches of "contain" were combined into a single fuzzy word match, four matches for "samples" and two for "sample" were combined into another single fuzzy match, and a number of other fuzzy word matches and semantic "clumps" occurred. Monitors placed by Semantics during processing of one-word theories detected coincidences between "samples" and "occur(ing)", between "contain" and "silicon", between "sample(s)" and "contain", and others. These events were ordered by their scores as assigned by the control component and the first two-word theory created was for "samples occur(ing)" (theory #21). The second two-word theory was for "sample(s) contain" (theory #22) and the third for "contain silicon" (theory #23). There was also a theory for "sample(s)" and the other word match for "contain" (theory #25). Theory #22 ("sample(s) contain") detected the match for "silicon" and produced theory #26 ("sample(s) contain silicon"). Also theory #23 ("contain silicon") detected the word match for "sample(s)", but it refrained from creating a duplicate of theory #26 after detecting its presence. Theory #26 was then passed to Syntax for verification and further prediction.

The word matches for theory #26 form a contiguous sequence of words from position 6 in the signal (60 ms from the beginning of the utterance) to the end, and Syntax was able to parse this sequence without knowing the word matches which occurred at the beginning of the sentence. After parsing the words that it was given, Syntax noticed word matches already in the word lattice for "many" and "any" ending at position 6 and proposed "much" and "there" and syntactic classes DET (determiner) and PREP (preposition), all ending at position 6. It also set monitors at position 6 looking for the classes ADJ, ORD, DET, N, V, NEG, and PREP.

The notice for "any" from Syntax for theory #26 resulted in a new theory for "any samples contain silicon" (theory #30), which detected the word "give" to its left. However, Syntax rejected "give any samples contain silicon" as being ungrammatical. The notice for "many" combined with theory #26 to give theory #31 ("many samples contain silicon"), which in turn noticed several words ending at the left end of "many" including the word "how". The scores of the words and the strategies applied by Control are such that the 38th theory formed was the complete analysis "how many samples contain silicon".

In the process of this computation, Semantics had placed 48 monitors of various types on specific words and concepts in the semantic network. There were 48 events (resulting from notices from monitors) left unprocessed on the event queue and an unknown number of potential events which could have been noticed if processing were continued. Syntax had created 104 configurations and 142 transitions in its internal syntax tables and set 51 monitors on positions in the word lattice.

Notice that the potential search space is vast, and the control mechanism is set up to systematically cover the entire space (if necessary) looking for an interpretation of the utterance. However, the order of processing theories is such that we have found the correct analysis at a very early stage of the search, leaving the vast majority of the computations on other paths undone.

H. Future Developments

As a consequence of further experience with the gradually evolving SPEECHLIS and further thought on the matter, it is clear that we could benefit greatly from a component presumably not used by Klatt and Stevens in their experiment. This is a prosodic component which knows the required relationships between syntactic structure and meaning, on the one hand, and the intonation contour and stress patterns of a speech utterance, on the other. When one considers the inherent ambiguity of the speech utterance which is entailed by the loss of word and phoneme boundaries and the relative uncertainty of identification of the elementary units of phonetic "spelling", and when one contrasts this with the fact that sentences read aloud are capable of resolving syntactic ambiguities which are not resolvable in written form, it is clear that some additional information must be present in the spoken utterance beyond a mere sequence of vaguely blurred sounds. It appears that this additional information is provided in the subtle variations in pitch, energy, and segment duration which are present in the spoken utterance and which seemingly relate the speech signal directly to the syntactic structure of the utterance. Although not presently a part of SPEECHLIS, we plan to include such a component in the system in the near future. It is anticipated that such information will greatly reduce the number of possible syntactic analysis paths which must be considered in the current system.

Another development planned for the future, and on which we are now working, is a much more sophisticated word verification component. This component will take a word match proposed by lexical retrieval or other sources, which has passed the tests of the current word matching component, and will perform a type of analysis-by-synthesis derivation of the detailed behavior of formants, transitions, etc. This will then be compared against the acoustic analysis parameters of the speech signal to obtain a more reliable word match score than that currently obtained. We expect this component to greatly reduce the number of accidental word matches accepted for consideration by the higher level components.

I. Conclusions

We have presented a brief overview of the various components of the BBN speech understanding system as of November 1973 together with a motivation for the structure of the system, the required capabilities of the individual components, and a brief description of how they work. More detailed descriptions of the individual components are contained in subsequent sections. The components of the current system are but crude approximations to their eventual forms, but they have been assembled into a total system in their current state in order to study their interactions. We believe that the development of the individual components will be more effective and the results more realistic if their development is done in the context of a total system rather than in isolation, and our experience so far bears this out. The project is now in a state where, for example, the interaction between the people working on acoustic analysis and those working on lexical retrieval and word matching as they try to make their components fit together has resulted in improvements to both sides, and this appears to be a continuing process.

A central issue of the BBN speech project is to gain insight into the ways in which the higher level linguistic components interact with the acoustic-phonetic and phonological components in the overall speech understanding process and to develop techniques for making this happen efficiently. We are especially concerned with discovering techniques which will be capable of dealing with a large vocabulary, a fluent English syntax, and a diversified range of semantic concepts, rather than attempting to optimize performance for small vocabularies and restricted syntax and semantics. We are concerned with finding the limits where increased vocabulary size, increased fluency of language, and increased range of semantic diversity cannot be handled by increased reliability in acoustic-phonetic and phonological analysis and word verification. Although the current capabilities of our system are but suggestive promises of what is to come, we think that the behavior of this minimal system on test sentences amply illustrates the potential power of the techniques which we have described. The full assessment of their capabilities must however await further development and testing.

II. THE ACOUSTIC/PHONETIC RECOGNITION PROGRAM

A. Introduction

Work on acoustic/phonetic recognition (APR) for automatic speech understanding has been going on at BBN for the past 3 years. Its state, as of November 1973, is well described in the paper "Where the Phonemes Are", presented at the IEEE Workshop on Speech Recognition in April, 1974, and included as Appendix C of this report. Familiarity with that appendix is assumed below, especially as it relates to the terminology used. In the past year we have been considering the inadequacies of that APR program and methods of eliminating them. Below, we list some of these inadequacies and the techniques which caused them. Spectrogram and parameter reading experiments and plans for the new APR under development are then discussed. Finally, we describe a statistics program which is being used to speed further development of the APR.

B. Problems With Old Methods

1. Segmentation

In the November 1973 system, the initial process of looking for possible phoneme boundaries (segmentation) depended mostly on the existence of abrupt changes in one or more of the acoustic parameters. Accordingly, the program was very good at locating boundaries manifested by rapid spectral changes as are found in obstruent-sonorant transitions. On the other hand, the shape or time evolution of the parameters was not fully used, causing slow transitions within sonorant sequences to be either missed entirely, misplaced or misinterpreted.

Secondly, the segmentation process was almost completely ignorant of acoustic/phonetic knowledge concerning the types of boundaries likely or even possible within a given region. This knowledge depends on the type of speech sounds which occupy the region. For example, one should not look for stop bursts or frication noise within sonorants.

Thirdly, confidence measures used in selecting boundaries were ad hoc. Confidences assigned to each analysis frame (every 10 msec) were used to determine which of several adjacent frames was a boundary. Then, the confidences on the boundaries (equal to the confidence on the frame at that point) were used to designate some boundaries as optional. These errors in confidences often resulted in incorrect segmentation or misplaced boundaries. Also, the confidences were not reliable enough to be used as an adjustment to the score in the word matching procedure.

Finally, the structure and demands of the program were so rigid that it was difficult to make its different sources of knowledge compatible. For example, even though the dip detector (which examines the energy in the preemphasized signal, ROD) found most of the correct boundaries by itself, the structure of the data and the program made it hard to incorporate new boundary information.

2. Labeling

In addition to the above inadequacies in the segmentation process, there were also inadequacies in labeling. First, the labeling routines usually took into account only the averages of some relevant acoustic parameters over the central half of the segment being labeled. This is sufficient for rough characterization, but for more precision, one must use the information in the shapes of the parameter tracks as well. For example, though the average energy level during vowels and nasals is not significantly different, vowels usually form energy peaks while nasals form energy dips. In other words, by using the average second derivative of the energy function, which is usually negative for vowels and positive for nasals, one can distinguish between these two classes of sounds.

Secondly, almost all information used in labeling was context independent. This caused many problems where there were large contextual effects (as near [r], [l] or silences). Experience here and elsewhere has shown that, in many instances, transitional cues contain much information which can aid in labeling. Also, boundary locations were computed independent of context. Since the labeling procedure is highly dependent on the location of the boundaries, this caused unnecessary labeling errors.

The decision procedure for each feature (examples of features are: voiced/unvoiced, sonorant/obstruent, nasal/vocalic, labial/dental/velar, etc.) consisted mainly of adding partial scores based on several acoustic parameters. Since each of these scores and the method for combining them was ad hoc., the resulting scores were not good measures of the likelihood of each feature. Since the set of phoneme labels was determined by the set of features with the highest scores, this procedure often resulted in incorrect answers.

C. Research

1. Spectrogram Reading

In order to get a better handle on the features of the spectrum which are important for recognition, we felt that it would be valuable to "read" several unknown spectrograms ourselves. Spectrograms were generated for sentences composed of a random selection of English words spoken in normal declarative sentence intonation. The purpose of the random selection was to eliminate syntactic and semantic information. Each of the readers independently attempted to segment and label the resulting utterances. Our reasons for making particular choices were then discussed. We then attempted to find words which matched the transcriptions. For those regions not matched by words, the person who knew the correct answer proposed words which fit the transcription roughly, but were incorrect otherwise. Reasons for rejecting the words were discussed. As was found in the experiment performed by Klatt and Stevens [21], we were quite good at rejecting incorrect word propcsals.

2. Parameter Reading

Since the computer will be segmenting and labeling from parameters, we decided to do a similar experiment using plots of the acoustic parameters available. This task was harder because we were now trying to correlate several one dimensional parameters, instead of looking at a single two dimensional picture. We found that we were able to segment and label fairly accurately with very few parameters, using the pole plots to determine formant positions. We felt that what we were looking at most was "significant" dips in certain parameters and the depth of these dips. We implemented the preliminary stage of this segmentation to see whether our hand techniques could be carried over to the machine and found that the algorithm did as well as we did on this limited task. We felt that these controlled parameter reading sessions greatly aided us in designing the segmentation and labeling program.

D. Solutions to Problems

1. Multiple Passes

Because the acoustic characteristics of a phoneme vary greatly with its context, it is very helpful to be aware of the nature of that context when making any decision as to its existence or identity. Therefore, we propose a multi-pass APR procedure which brings context into the segmentation and labeling process. Each pass consists of four steps: initial segmentation, initial labeling, adjustment of boundaries, and relabeling. Boundaries are adjusted so that they correspond to reliable acoustic events which are determined by the results of the initial labeling. Relabeling is then performed using the adjusted boundary times. Each pass operates on regions generated by the segmentation in the previous pass, performing more detailed segmentation and labeling that use more detailed contextual information. Our current plan calls for a three-pass APR procedure, as follows:

(a) Find "obvious" boundaries between sonorant and obstruent regions. This can be done primarily using the energy in the low frequencies.

(b) Divide sonorant regions further into vowel and non-vowel regions by looking for dips in mid and high frequency energy. Also, divide obstruent regions into frication and stop regions.

(c) Some of the regions generated by the first two passes contain more than one phoneme. Accordingly, within each region, boundaries are detected using region-specific parameters and routines. For example, if the region is vocalic, formants are used in addition to the other parameters. Each segment in the resulting segment lattice is then labeled using the partial results for the adjacent segments.

This multi-pass approach assures maximal use of robust, detectable contextual information.

2. Reliable Boundary Confidences

The confidence associated with each boundary reflects, to some extent, both the reliability of a cue in signalling a boundary and the strength of the cue. There are several cues used in this program for finding boundaries. The program searches for dips in some parameters, rapid transitions in others, formant motion in vocalic sequences, etc. In order to compute a confidence on each boundary, a parameter relevant to the evidence of a boundary should be used. For instance, the depth of a dip is a good indicator of the reliability of that dip as a boundary. We propose to determine these relationships statistically so that the confidences given will be meaningful when used to compute the score on a word match.

3. Context Dependency

In using context when labeling a segment it would be very helpful to know, with absolute certainty, the identity of the adjacent segments. However, if context is used, then incorrect hypotheses about the identity of the adjacent segments could lead to labeling errors. In those cases where these hypotheses are likely to be incorrect, it would be advantageous to consider all possible relevant contexts, and compute different results for each postulated context. For example, one way to decide between [p,t,k] is to look at the 2nd and 3rd formants in the following vowel. The formants typically "point" to a frequency (locus) which is characteristic of the place of articulation of the plosive. However in the case of [k], this locus frequency depends on whether the following vowel is rounded or not. Since the following vowel is not always reliably determined, one must consider two allophones of [k]; one followed by rounded vowels, the other followed by unrounded vowels. (An allophone is one of the variant forms of a phoneme, i.e. the aspirated [p] of "pit" and the unaspirated [p] of "spit" are allophones of the English phoneme [p].) Then the score on [k-rounded], for example, is the probability that the relevant acoustic parameters (voice onset time, burst spectrum, formant motions, etc.) would have the values they do, given that it is a [k] and the following vowel is rounded. When used in word matching, the roundedness of the following vowel is known and only the single appropriate allophone of [k] need be considered. Of course, one wants to minimize the number of different allophones that need to be considered, but a reasonable balance can result in a large improvement in word matching.

4. Probabilistic Labeling

Word pronunciations will be modeled as allophone sequences. While the APR does not have access to the word pronunciation models, the word matcher does. Consequently - in an effort to provide the word matcher with the maximum amount of relevant information about each segment - a labeling philosophy to directly characterize each segment probabilistically has been adopted. This is contrasted with the philosophy of explicitly labeling each segment as a single allophone.

These two philosophies differ in a way which may not be immediately evident to the reader. In either case the word matcher (which knows pronunciation models as allophone sequences) needs a score for every allophone it matches with each segment. The matching score is the probability that this allophone, when spoken, would have resulted in the observed acoustic characterization.

In the first case, although the APR provides these scores directly, there are really two processing steps involved. First, parameters thought to be relevant to the recognition of the segment are designated as the observed acoustic characterization. Then, probability distributions (one for each allophone) which depend on these parameters are evaluated to produce scores for the different allophones. The specific values of the parameters observed in each segment are used in these evaluations. The segment characterization produced by the

APR (and presented to the word matcher) is a vector of computed scores (probabilities) with one element per allophone.

In the second case the APR provides only a single label, which can be thought of as its observed acoustic characterization. In this case, however, an interface between the APR and the word matcher effectively provides the desired scores by consulting a confusion matrix which contains probabilities for every combination of allophone and segment label. As long as variations in the relevant acoustic parameters do not cause a segment label change, none of the scores provided to the word matcher by the interface will change. However, this is contrary to the observation that variations of acoustic parameters for a single phoneme do in fact change the confusion likelihood of that phoneme with other phonemes.

The first philosophy results in a better characterization of the segment because relevant parameter variations otherwise ignored (e.g. whenever the parameter variations would not have caused a segment label change) can be incorporated in the word matcher scoring mechanism. Since this technique requires evaluating all possibilities, it is more costly, however. Therefore, what we have chosen is a combination of the two techniques. For those phonemes which are very unlikely to match a particular segment, the probabilities predicted by a long term confusion matrix are a good approximation to the likelihoods which would be computed explicitly. For example, if one believes a segment to be a [t], the probability distributions for [t,p,k,d,n] should be evaluated using the observed parameters. But the scores on each of the vowels are all bad, so they will be fairly insensitive to this particular manifestation of [t]. This means that not all scores in the vector need be computed for every phoneme label on each segment; most can come from the confusion matrix, while those that are sensitive to parameter variations will be computed individually.

5. Speaker Normalization

The current APR does not employ speaker normalization to any great extent. While minimum and maximum values of the first three formants can be supplied in order to aid formant tracking, it was not found to make a major improvement. Instead of recording a set of vowels to determine the speaker's vowel formant space, the vowel classifier normalizes the observed formant frequencies based on the average of the pitch fundamental frequency, and then compares these "self normalized" formants to a universal vowel table which is used for men, women, and children.

It is hoped that most algorithms in the APR under development will be speaker independent. This can be facilitated by the use of relative, rather than absolute thresholds. (For example, using the depth of a dip in energy instead of the minimum value during the dip.) Areas where normalization may be necessary or helpful include: specifying frication spectra during fricatives and plosives, and accounting for dialect-based effects.

E. Statistics Program

An interactive statistics package has been developed which permits the user to perform various acoustic/phonetic experiments. These allow him to approximate the probability distribution of a particular value of an acoustic parameter, given that a particular feature was present. The user specifies the phonetic context in which he is interested, in terms of phonemes, features, stress markings, word or syllable boundaries (required, allowed, or disallowed), orthographic spellings, or any combination of the above. An experiment then, is defined by supplying a series of simple functions which are to be evaluated each time the specified context is found. Functions can range from simple arithmetic or Boolean operations to complicated valley searching procedures. The program prompts the user for functions and arguments. A typical protocol for a function specification is shown below, with the responses of the user underlined:

(The function will find the last frame between the centers of segments 1 and 2 in the required context in which the derivative of the parameter ROD is greater than 2.0.)

Function: next time

Parameter: Derivative of parameter: ROD,

From: center of segment #: 2

Until: center of segment #: 1

is greater than 2 considering: only absolute values.

All arguments can be the results of previous functions. The user then supplies a list of names of utterances from the data base, or a set of criteria for choosing utterances to consider. These criteria include speaker, sentence number, token number, sex of speaker, date of recording, sampling rate, speaking mode, subject domain, etc. Any of the criteria may be left unspecified.

Results can be examined at any desired level, from a complete listing of each occurrence and all partial results of the experiment, to interrogating the program for the minimum, maximum, average, or a complete listing of all the values of any of the partial or final results. The user can also obtain a graphic display of a histogram, density distribution, cumulative distribution or scatter diagram in two or three dimensions.

All interactions are under user control, with verbose prompting from the program. Any partial state can be temporarily saved on a file and updated later. Results of two or more complementary experiments (e.g. one on voiced plosives and another on unvoiced plosives) can be superimposed to provide an intuitive feel for the usefulness of an algorithm. This program has already been used successfully in testing and improving some labeling algorithms.

III. LEXICAL RETRIEVAL

A. Introduction

Automatic speech understanding requires the development of programs which can formulate hypotheses about the content of an utterance and attempt to verify them. One example of such activity in the BBN Speech Understanding System (SPEECHLIS) is both the top-down and the bottom-up formulation of hypotheses about the particular words which occur in an utterance and their subsequent verification against a completed feature analysis of the utterance. It is at this interface between acoustic transcription and word matches that knowledge about the vocabulary, phonemic spellings, phoneme similarity, and phonological rules is represented and applied.

Lexical retrieval in SPEECHLIS then comprises both data-driven hypothesis formulation and word verification. The scope of SPEECHLIS makes both abilities vital. For task domains which deal with a small vocabulary and/or have strong syntactic and semantic constraints, the number of words which could appear in a given region of the utterance can be limited substantially. In such systems, one can list the words and word sequences allowable at a given point before considering the acoustic transcription, match them against the acoustic transcription, and then order them on the basis of match quality. The BBN speech understanding project on the other hand has chosen to develop a system for tasks in which such higher-level constraints are not strong enough to radically limit the set of possible words in early stages of the understanding process. Instead, information from the acoustic transcription itself must be used in an initial pnase of hypothesis formation to suggest words which match well. These words then suggest to higher-level knowledge sources other words which might occur in their context and which are subsequently matched and verified against the data.

Lexical retrieval occurs in SPEECHLIS at the interface between acoustic-phonetic recognition programs which construct the acoustic transcription, and syntactic, semantic, pragmatic, and control programs which combine word matches into tentative hypotheses about the structure and meaning of the utterance. The lexical retrieval programs have two tasks: to use the acoustic transcription to propose words for which acoustic evidence exists (Lexical Proposal), and to evaluate how well a proposed word matches the acoustic information (Lexical Matching).

In this chapter we describe the way in which Lexical Retrieval fits into the November 1973 SPEECHLIS system, with regard to the strategies for Lexical Proposal and Lexical Matching and the representation and use of phonological rules. We then describe subsequent work on a new lexical verification subsystem which matches word-spellings or word-sequence-spellings against a parametric representation of the utterance as opposed to the acoustic-phonetic transcription. This subsystem has not yet been integrated into SPEECHLIS. Finally, some longer-range work in phonology is briefly described.

B. Lexical Retrieval in SPEECHLIS

1. Data Structures

The lexical retrieval programs have access to data structures which represent the acoustic transcription of the utterance, the vocabulary, a corpus of phonological rules, and a "phoneme similarity matrix".

a. The Acoustic Transcription

The acoustic transcription is in the form of a structured collection of SEGMENT descriptors. By a segment we mean a portion of the utterance which is hypothesized to be a single phoneme. Each segment has a description which could in principle specify the phonemic identity of the segment, but in general merely constrains this identity to one of several phonemes. This set of phonemes represents the acoustic features that were detected in a feature analysis of the segment. The number of phonemes in the set reflects the level of detail in the result of the feature analysis. This level of detail is adjusted for each segment to maintain a reasonable balance between vagueness of feature description and confidence that the feature description is correct. For each segment and for each boundary between segments in the segment lattice, a crude measure of this confidence is represented. Alternative hypothesized segments may overlap in the utterance, resulting in a lattice of segment descriptors rather than a single string. Figure 1 gives an example of such a SEGMENT LATTICE. The numbers along the top are used to identify the boundaries between segments. Each segment is labeled with its set of alternative phonemes. This structure allows for the representation of uncertainty or ambiguity both in the determination of the segment boundaries and in the identity of a segment.

b. The Vocabulary

Each of the words in the vocabulary (approximately 250 in the lunar rocks domain) has a set of its most likely pronunciations given as lists of phonemes and syllable boundary markers. On the average, there are about 2 pronunciations represented for each word in the vocabulary. Associated with each phoneme is an estimate of the probability that it will be deleted when the word is actually pronounced. Associated with each vowel is an expected stress value (either "primary stress", "secondary stress", or "unstressed"). There also exists a cross-referenced data structure for the vocabulary which has for each phoneme a list of words which either start or end with that phoneme, and for each ordered pair of phonemes a list of words in which that phoneme pair occurs, with the associated indices into the phonemic spellings.

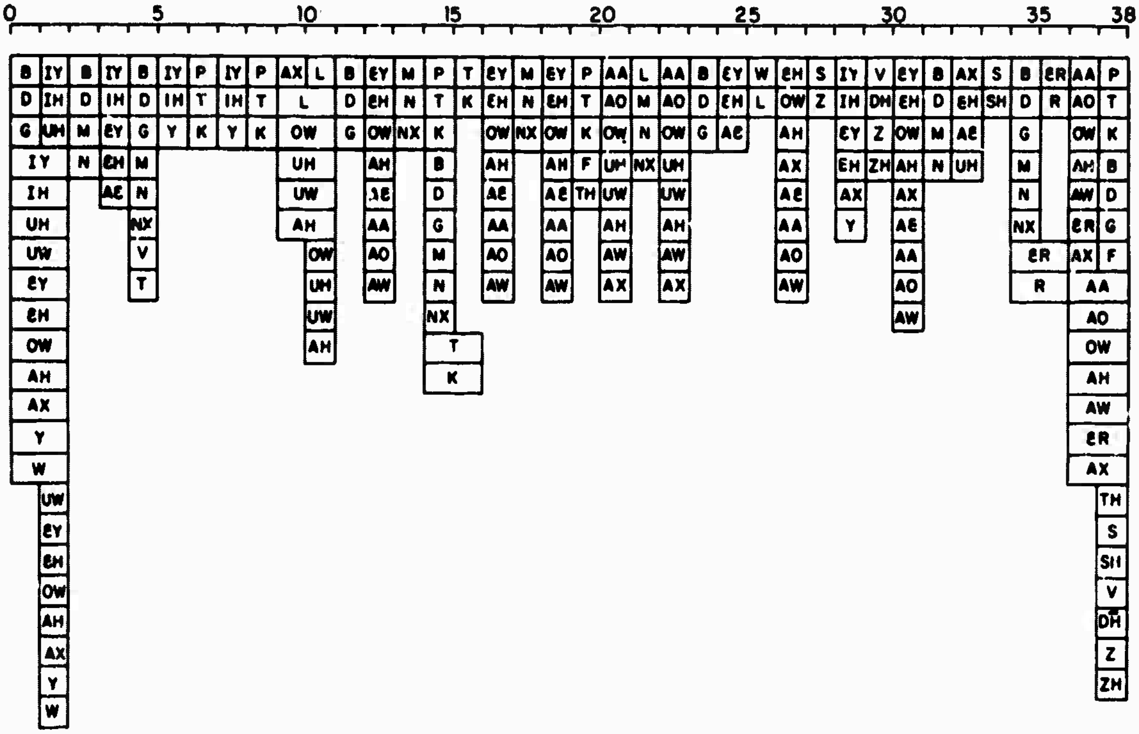


Figure 1. Segment Lattice

c. The Similarity Matrix

Information about the similarity of phonemes is represented in a SIMILARITY MATRIX. Each entry in this matrix is an estimate of the likelihood for a pair of phonemes (P1 P2) that a segment labeled P2 is really P1, i.e. how "similar" is P2 to P1. The similarity matrix has two uses: to adjust for the known performance of the acoustic-phonetic programs, and to account (crudely) for variations in phoneme pronunciation that are not yet implemented as phonological rules. In the present system, these estimates are derived from our intuitions; as we gather statistics from real instances of phoneme confusion, we will adjust these estimates.

d. Phonological Knowledge

Phonological knowledge tells us about the ways in which the pronunciation of words can vary. One of the tasks of the lexical retrieval programs s to take account of such knowledge as they look for word matches in the segment lattice. In addition to the phonological information in the phonemic dictionary and in the similarity matrix, SPEECHLIS has a corpus of context-dependent analytic phonological rules. These are represented in a collection of data structures which specify contexts in the segment lattice in which phonemes can be changed, inserted, or deleted. Because they represent transformations from observed phonetic sequences to sequences which conform to the phonemic spellings in the dictionary, these are termed <u>analytic</u> (as opposed to generative) phonological rules. Each rule has three components:

(1) A template describing the necessary context to be sought in the segment lattice.

(2) A description of a new branch to be added to the lattice, given the presence of the necessary context. The attributes of this new branch can depend on the attributes of the context found in the lattice.

(3) A predicate (see below).

The segment lattice as constructed by the acoustic-phonetic programs represents initial (and currently, largely context-free) hypotheses as to the existence of boundaries and acoustic features of segments in the utterance. After this segment lattice is constructed, a rule-interpretation program applies the set of rules to the lattice. The action of these rules is never to change the existing lattice structure, but rather to add new branches which specify optional paths through the lattice. In general, the admissibility of a new branch cannot be entirely determined from the information in the lattice alone. It is the job of the predicate to complete the task of determining the applicability of the rule when a portion of a particular phonemic spelling is being considered by the lexical matcher.

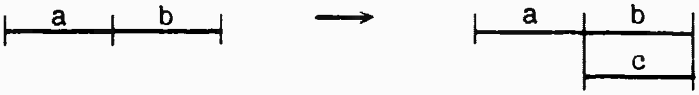
When the lexical matcher finds a path through the lattice which is an acceptable match for a particular lexical entry, it examines the segments in that path for predicate function pointers. For each such pointer that it encounters, it calls the predicate function, giving as arguments the phonemic spelling of that lexical entry, the position within that spelling, and a pointer to the segment in the lattice. The predicate function, which can be an arbitrary piece of code, performs a computation on these arguments and returns <u>true</u> if it accepts the use of the segment in that word match or <u>false</u> if it rejects it. (A possible generalization would be for the predicate function to return a confidence measure. However the evaluation mechanism in the current word matcher does not seem sophisticated enough to warrant this.)

Although a rule which adds a branch to the segment lattice, based on existing structure, is analytic, the condition imposed by the predicate function associated with the branch is a function of the underlying form in the lexicon, giving the applied-rule-plus-predicate a generative flavor as well. These predicate functions can be used in three ways:

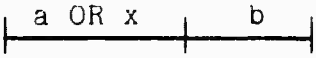
(1) To check a context condition not checked in the "analytic" application of the rule, because relevant factors may not be available in the segment lattice. These factors include:

(a) Stress
(b) Place of articulation
(c) Position of segment with respect to word boundary

(2) To compensate for "sloppiness" in the context of the "analytic" application of the rule. For example, if the rule were:



and the segment lattice were labeled:



where *x* is some set of labels which does not fit the description *a*, then if the segment *c* were to be added, an unwanted path *x*-*c* would exist in the augmented lattice. One way to eliminate this would be to bridge the entire context by a two-segment branch consisting of *a* followed by *c*. This partial copying can become quite complex in general and it can result in duplication of much of the lattice. Instead, the segment c is added anyway, but any word matches using the unwanted path are summarily rejected by the predicate function.

(3) A rule of general usefulness may fail to apply for a few exceptional words. Such exceptions may be detected in a predicate function.

Additional branches inserted by the rules ensure that the lexical retrieval programs will consider those standard word spellings which could have the indicated phonological variation. Such a scheme serves to select for consideration variations on the standard phonemic spelling ONLY WHEN the standard spelling is not represented in the segment lattice AND a variation of it is possible on the basis of the detection of an appropriate context (in the segment lattice) for the application of the phonological rule. Furthermore, the pattern match processing necessary to detect such contexts for determining the applicability of each phonological rule is done only once in a special scan over the segment lattice; it is not necessary to analyze the segment lattice anew for applicable phonological patterns each time a new word is considered by the lexical matcher.

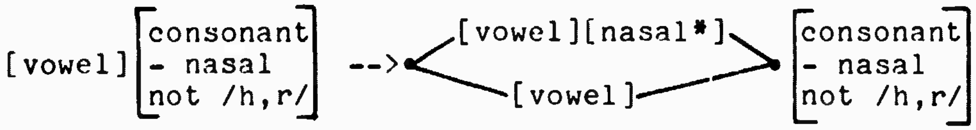
Example: Nasal Deletion Rule

Generative form:

$$\begin{bmatrix}\text{consonant}\\ +\ \text{nasal}\\ \text{place}\end{bmatrix} \text{ --> } 0 \Big/ [\text{vowel}] \_\_\_ \begin{bmatrix}\text{consonant}\\ -\ \text{nasal}\\ \text{place}\\ \text{not } /\text{h,r}/\end{bmatrix}$$

"A nasal consonant is deleted if it occurs immediately after a vowel and immediately before a nonnasal consonant (not /h/ or /r/) with the same place of articulation."

Analytic form:



* Predicate function requires:
1. Nasal not word-initial.
2. Preceding segment must be a vowel.
3. Nasal may be word-final (if it is, predicate has no way of checking the following segment)
   OR
   Following segment must be a nonnasal consonant (not /h/ or /r/) with same place of articulation as the nasal.

"If there exists a path through the lattice such that a vowel segment is followed by a nonnasal consonant (not /h/ or /r/), then bridge the vowel segment by a two-segment branch consisting of the vowel followed by a nasal. Attach a predicate (described above) to the nasal segment." (If such a branch bridging the vowel already exists, then no new branch need be added.)

The phonological rules component is implemented as a set of BCPL functions which live in the lexical retrieval fork. The rules themselves are elementary data structures describing the necessary context for the rule to apply and each segment of the new branch to be added to the lattice. The properties of these new segments can be expressed absolutely (e.g., duration = 30 msec) or relative to some segment in the context (e.g., duration = 80% of the first segment of the context, or stress = 1 lower than that of the third segment). The predicate functions may be arbitrary, but in practice they mainly call a small set of functions which check segment descriptions and vowel stress.

The actual program fragment which specifies the Nasal Deletion Rule is given below. It consists of three parts - a set of phoneme cluster definitions (which are used to describe segments), the rule, and its predicate. The notation for expressing the rule is far from a linguist's notation, but it is quite straightforward. The example is illustrative, not exhaustive.

```
//Definitions of phoneme clusters, used in the rules and
//in the predicates.
static
{ VOWEL:=table 0,12,UW,UH,OW,AO,AA,AH,AE,EH,IH,IY,AX,EY
  CONSONANTNOTNASALHR:=table 0,14,P,T,K,B,D,G,F,V,TH,DH,S,Z,SH,ZH
  NASAL:=table 0,3,M,N,NX
  phM:=table 0,1,M
  phN:=table 0,1,N
  phNX:=table 0,1,NX
  LABIALNONNASAL:=table 0,4,P,B,F,V
  DENTPALNONNASAL:=table 0,8,T,D,S,Z,SH,ZH,TH,DH
  VELARNONNASAL:=table 0,2,K,G
}

//The Deleted Nasal Rule itself consists of 3 parts:
//  Description of the necessary context
//  Description of the new branch to be added
//  A string giving the name of the rule
let DeletedNasal:=list
  (list 2,                      //The context has 2 segments,
    OPERAND,,VOWEL,                //a vowel followed by
    CONTEXT,,CONSONANTNOTNASALHR), //a nonnasal consonant
  (list 2,                      //The new branch has 2 segments:
    (list PHINTERSECTION,,1,       //intersection with the VOWEL
      RDURATION,,80,               //duration=80% of the VOWEL
      RCONFIDENCE+1,,100,          //confidence=100% of the VOWEL
      RSTRESS+1,,0,                //stress=same as the VOWEL
      RBCONFIDENCE,,60,            //right bdry confidence=60
      ENDLIST),
    (list PH,,NASAL,               //The 2nd segment is a nasal
      CONFIDENCE,,100,             //100 means exact match only
      STRING,,"DeNasal",
      PREDICATE,,DeletedNasalPred, //predicate on this segment
      ENDLIST)),
  "DeletedNasal"                //String giving rule name

//The predicate function for this rule:
and DeletedNasalPred(spellingindex,segptr):=valof
{ let oldrsw:=rightsw
  predspx:=spellingindex
//check that the preceding segment is a vowel, and that the
//nasal and following consonant have same place of articulation
  let yesno:=check(-1,VOWEL,false)&
        ((check(0,phM)&check(1,LABIALNONNASAL,true))\
         (check(0,phN)&check(1,DENTPALNONNASAL,true))\
         (check(0,phNX)&check(1,VELARNONNASAL,true)))
  if traceflag do tracepred(yesno,segptr,"DeletedNasal")
  rightsw:=oldrsw
  let cnt:=lv DeletedNasal!(yesno->NACCEPT,NREJECT)
  rv cnt:=1+rv cnt
  resultis yesno
}
```

The function which applies such rules to a segment lattice takes as input an ordered list of the rules. Each rule is applied from left to right across the lattice before proceeding to the next rule, but rule repetition may be accomplished by including a rule name in the list more than once. Statistics are accumulated on how many times each rule is applied and on how many times its predicate function returns _true_ and _false_. If a trace flag is enabled, each rule application and each predicate function execution is described on an output file, which may be the user´s terminal.

The 11 rules now implemented are enumerated below. Of these, four of them are "real" phonological rules (such as the Deleted Nasal Rule described above), and seven account for other phenomena which are more appropriate to the segmenter/labeler component, but which can be expressed and applied in the same format as the phonological rules (such as the InitialVowel rule). Their order of application is the same as the order in which they are listed below; the only crucial ordering is that DeletedNasal follows FinalVowel.

(1) SyllabicLMN1: An L or nasal appearing between two consonants, the first of which must not be R, may be a segment which was originally preceded by a schwa, but which is now syllabic, the schwa having been deleted. Insert such a two-segment branch bridging the L-or-nasal. The predicate requires that neither the schwa nor the L-or-nasal may be word-initial. (E.g., "people" [P IY P L] --> [P IY P AX L])

(2) ConsolidatePlosive: A plosive segment followed by an unvoiced segment may be an unvoiced plosive with such a long enough aspiration interval that the aspiration gets labeled as a separate segment. Bridge the pair with an unvoiced plosive. No predicate is necessary. (Since the current acoustic-phonetic recognizer does not attempt to identify place of articulation in plosives, this form suffices. It would be natural to make the added unvoiced plosive segment have the same place(s) of articulation as the plosive it bridges. This phenomenon is most probable when the second segment is followed by a stressed vowel, with a possible intervening W, R, Y, or L. However, the acoustic-phonetic recognizer currently makes this mistake sufficiently often that this more stringent condition is omitted for now.)

(3) FinalVowel: A vowel followed by a silent segment (e.g., utterance-final) may have an undetected weak consonant (P, T, K, B, D, G, F, TH) after the vowel, so insert (an optional) one. The predicate checks that the first segment is indeed a vowel.

(4) DeletedDH: A nasal or fricative (but not DH) segment followed by a vowel may have resulted from the deletion of a word-initial DH, so insert an optional DH. The predicate requires the DH to be word-initial. (E.g., "in the" [IH N AX] --> [IH N DH AX])

(5) DeletedNasal: described above.

(6) InitialVowel: A silent segment followed by a vowel (e.g., utterance-initial) may have an undetected weak consonant (P, B, D, G, HH, F, TH) preceding the vowel, so insert (an optional) segment so labeled. Predicate checks that the 2nd segment is indeed a vowel.

(7) InitialR: like InitialVowel, but adds P, T, K, B, D, G, F, TH.

(8) InitialL: like InitialVowel, but adds only P, K, B, G, F.

(9) InitialFricAsp: A silent segment followed by a fricative or aspiration segment may instead be a plosive, so insert a plosive branch across the frication/aspiration segment. No predicate.

(10) FinalS: like FinalVowel, but adds only P, T, K.

(11) FinalNasal: like FinalVowel, but adds only P, T, K, B, D, G, TH.

After applying these 11 rules to the initial segment lattice, we have seen it increase in size by factors of 2 to 3. The total number of word matches has increased by about the same factor. However, the number of <u>correct</u> words matched has also generally increased as a result of the application of the rules.

e. Output

The output of the lexical retrieval programs is a set of WORD MATCHES. Each word match is a correspondence between one phonemic spelling of a word and a path through the segment lattice. A score is associated with each word match to indicate how well the phonemic spelling matches the sequence of segment descriptors. Word matches of sufficient quality to be examined by Syntax, Semantics, and Pragmatics are entered into a WORD LATTICE (Figure 2). In this figure, for example, the word "mean", spelled [M IY N], matches from position 2 to position 5 in the lattice, while the word "print", spelled [P R IH N T], matches from 0 to 5. The first of the two numbers in parentheses for each word represents the score of the word match. The second number represents the maximum possible score for a word of its length (number of phonemes).

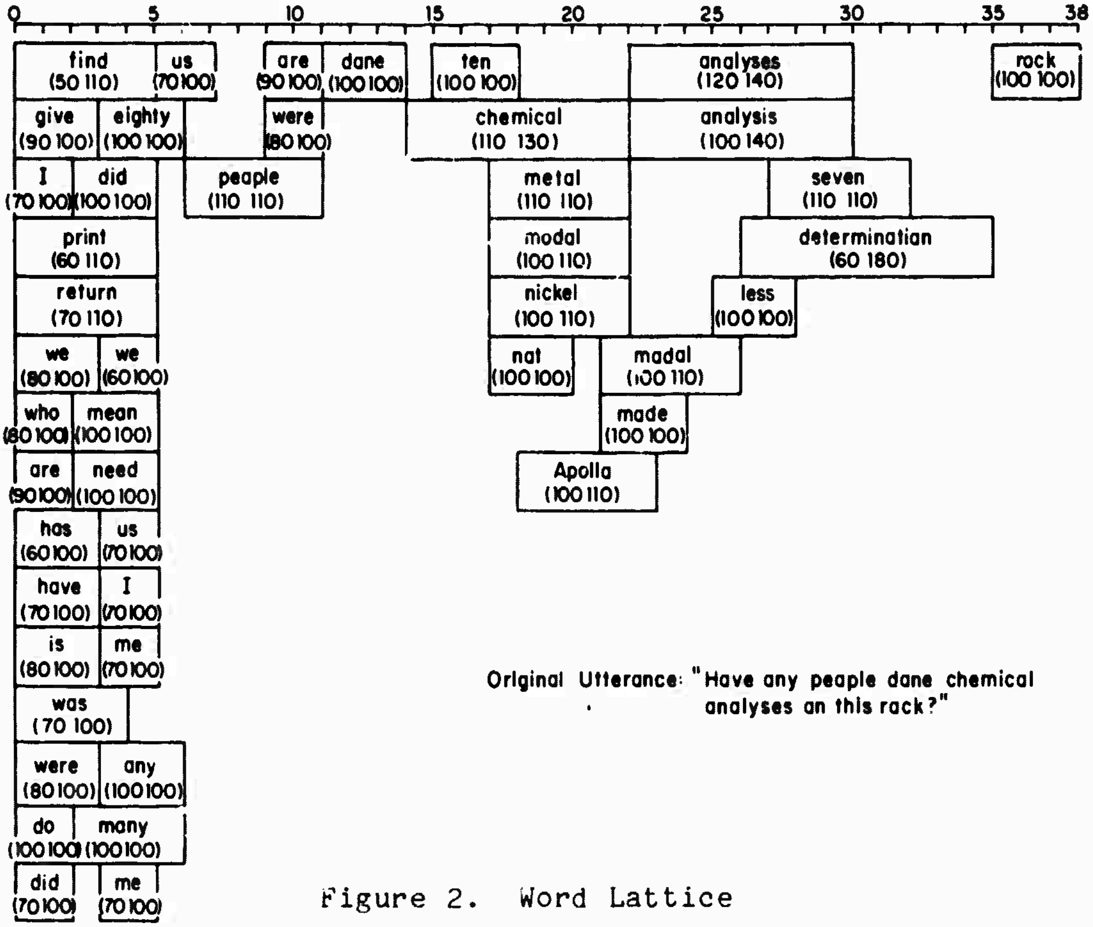


Figure 2. Word Lattice

2. Usage

The overall control strategy for SPEECHLIS starts from an acoustic transcription which has been expanded by the analytic phonological rules. Next a scan is performed over the entire segment lattice to find word matches anywhere in the utterance which are longer than two phonemes and which match well. These are used to construct an initial word lattice. Then some top-down hypothesizing occurs as likely sentence-initial words (i.e. question words, auxiliary verbs and imperative verbs) are matched at the beginning of the utterance. Any such word matches are added to the word lattice. The system then enters a phase of hypothesis formation, in which word matches from the word lattice are combined into word match aggregates (called THEORIES) on the basis of semantic, syntactic, or pragmatic justification. As the system attempts to verify, enlarge, and combine these theories, the lexical retrieval programs may be called upon to match words which have been proposed by Syntax, Semantics, and Pragmatics. Examples of such proposals are: content and function words which are likely to be adjacent to an existing word match and possible inflectional endings and auxiliary verbs for a given word.

An extensive set of parameters are available for controlling the activity of the lexical retrieval programs. These parameters allow the control component to specify, for example: 1) acceptable word lengths and word match quality; 2) either end point of the match; and/or 3) the region of the segment lattice in which the match is to be made. In addition, there are parameters for selecting one of several strategies for searching and matching, including the consideration of word matches with missing or extra segments. These strategies are described below.

C. Strategies

1. Lexical Proposal

There are two ways in which words can be suggested for consideration from the information in a specified region of the segment lattice. One way is to consider, for each phoneme of each segment in the region, the set of word spellings which begin or end with that phoneme. This is called an "anchored" scan. Alternatively, there is the "unanchored" scan, in which a word spelling is proposed if it has a specified pair of adjacent phonemes anywhere in its spelling. For each pair of adjacent segments in the specified region of the segment lattice, the set of such phoneme pairs is computed as the cross product of the phoneme sets labeling the segments. The unanchored method is currently being used in SPEECHLIS for the complete initial scan.

2. Lexical Matching

The lexical matching algorithm is a "recursive tree walk". For a given boundary in the segment lattice, a given phonemic spelling, and a given index to one of the phonemes in the phonemic spelling, this algorithm walks the segment lattice postulating phoneme-segment matches. The index into the phonemic spelling is "aligned" with the given boundary in the lattice. If the given index divides the phonemic spelling into two parts, as is usually the case during an unanchored scan, then a "middle-out" walk is performed. Otherwise, either a "left-to-right" or a "right-to-left" walk is done, depending on whether the index points to the first phoneme (left end) of the phonemic spelling or to the last phoneme (right end). For possible missing or extra segments and branch points in the segment lattice, the matcher is called recursively to consider the alternate paths through the segment lattice.

Each postulated phoneme-segment match is evaluated on the basis of the similarity between the given phoneme and the most similar phoneme in the segment label. The phoneme-segment match score is quantized as a number between zero and 5; the higher score represents a better match. Each phoneme-segment evaluation is used to adjust a cumulative overall word match score. This score is initialized to the maximum possible score for the word and is incrementally adjusted as phoneme-segment match scores are considered. This maximum score depends on the length of the phonemic spelling; longer words have a higher maximum.

For each vowel in the phonemic spelling, a simple analysis of the segment duration is used to adjust this word match score. This is done on the basis of whether the vowel is tense or lax, and whether it is stressed or unstressed in the word spelling. For example, the appearance of an unstressed, lax vowel in a segment having a duration greater than 100 milliseconds is assumed very unlikely. Any word match in which such a phoneme-segment match is a component will have its score decreased substantially If a missing or extra segment is postulated, its score is computed from a priori information (in the dictionary) about the likelihood of such a phenomenon for the indicated portion of the phonemic spelling.

If the word match score falls below a specified word match score acceptance threshold, consideration of this path through the segment lattice is terminated. Note that, because of branching in the segment lattice, it is possible for a phonemic spelling to match along more than one path through the same region of the segment lattice. Of these matches only the ones with the best scores are entered into the word lattice.

D. Performance and Future Work

Since the first version of SPEECHLIS has not been tested extensively, we are not yet able to present a thorough analysis of the lexical retrieval performance requirements for acceptable overall system performance. From the small set of utterances that we have tried using this system, however, we have formed some tentative impressions:

(1) For a normal-sized utterance (e.g., 9 words; 5 content words), the system will probably perform well with an initial word lattice having roughly 100 word matches, if all or all but one of the content words are present with good scores Note that function words are not expected to be found in the initial scan; rather they are looked for when explicitly proposed by the syntactic component of the system.

(2) The quality of overall system performance depends greatly on the quality of lexical retrieval performance. This in turn depends on two factors: the amount of information in the segment lattice, and the effectiveness of the lexical retrieval programs in utilizing that information. The payoff of improvements in either of these two areas will be high.

(3) Circumstances have precluded extensive testing of the analytic technique for implementing phonological rules and the 11 rules themselves, but some tentative conclusions can be made. This method does not seem well suited for implementing some types of phonological processes, especially deletion processes which destroy much or all of their triggering context. We will probably change to a system of generative rules which effectively expand the dictionary entries [2,4]. Many of the analytic rules will survive in some form, since rules something like, for example, the InitialVowel and ConsolidatePlosive rules should exist in the acoustic-phonetic recognition program, where they have access not only to the segment lattice but also to the parametric representation of the utterance.

Work underway to improve lexical retrieval performance is directed toward increasing the number and quality of correct word matches found, especially from the initial scan, while keeping both the number of incorrect word matches and the processing requirements within manageable limits.

To further develop our experience with and insight into lexical retrieval, we are gathering statistics on the relative reliability of different kinds of segments and boundaries in the acoustic transcription and, for each word in the vocabulary, the relative reliability of detecting those features and phonemes which one would expect to be "robust" (e.g., stressed vowels and strident fricatives). In the future, we expect to use such robust phenomena for word proposal, rather than the rather loose criteria described above.

One pressing problem is the need for a more rigorous foundation for computing word match scores. As we learn more about the relative reliability of parts of the acoustic transcription and about ways in which new correlations between phonemic spellings and acoustic features should be used to influence word match scoring, we will be able to improve our present (largely intuitive) techniques.

Since we are committed to dealing with larger vocabularies (1000 words and over), one of our goals is to develop lexical retrieval techniques which are efficient and effective and largely independent of vocabulary size. A new lexical retrieval component is under development which will satisfy this condition as well as providing a better foundation for word match scores. It will be described in subsequent reports.

1. Lexical Verification

Prior to December 1973, our system employed a bottom-up approach in creating a phonetic transcription (segment lattice) from the raw acoustic input. This segment lattice alone provided the data for both word proposals and word verification. This caused two major problems: there were far too many hypotheses generated, and errors or basic shortcomings in this domain were irrecoverable.

Given the results of the Klatt-Stevens spectrogram reading experiment [23], it seems clear that the ability to return to acoustic evidence for verifying word hypotheses is important to correct identification. This is because one can then verify the consistency of all acoustic clues with respect to the given word hypothesis. Assuming that phonological and coarticulation processes are best described by rules which are generative in nature, it seems that an analysis-by-synthesis procedure is needed to overcome inaccuracies in a strictly bottom-up phonetic analysis and to decode the effects of phonological rules.

We are therefore in the process of constructing a lexical verification component which will be able to function in an analogical manner. That is, given a generalized phonetic transcription of the candidate word sequence, consisting of a broad phonetic transcription, syllable boundaries and word boundaries, the synthesizer will transform it into a set of acoustic parameters for comparison with the acoustic parameterization of the unknown utterance. The degree to which the parameterizations are in some sense equivalent over a specific interval of the utterance gives a measure of likelihood for the hypothesis being correct.

A synthesis-by-rule program whose input consists of the above generalized transcription has been written. Based on a terminal analog model of speech production [20], it does a direct phonetic-to-acoustic parameter conversion using rules derived from relevant data collected from spectrograms or extracted automatically from digitized speech. The program's output parametric representation presently consists of three formant frequencies with segment durations.

Concurrently, a mapping strategy for comparing the synthesized parameters against the unknown utterance is under development. The strategy will take into account time registration, time and frequency normalization, and match score computation. Given a location and context for new word hypotheses, the portion of the unknown utterance over which matching is permitted will be restricted. The overall match score will be a composite of segment match scores which depend on pattern differences in the parameters relevant to each particular segment type.

As an aid in formulating scoring strategies, we did some informal experiments in spectrogram reading (mentioned earlier in Section II). People expert in this task were given spectrograms and asked to verify the presence of hypothesized words. The spectrograms consisted of random words spoken as continuous utterances so that only acoustic evidence and not syntactic and semantic relations would be used in judging the acceptability of word hypotheses. Deviations from what the the experts considered ideal exemplars were recorded and classified according to their severity. Preliminary results confirmed the importance of formant transitions and durations in making these judgments. It is also interesting to note that the experts tended to lock for features which could rule out rather than support a given hypothesis. Capturing these discriminations within a procedural framework is a primary goal of this research.

By synthesizing a more detailed description of the hypothesis, we hope to refine our scoring in cases where discrepancies are subtle and detailed analysis may be required. Additional parameters which might be used for word verification (based on their perceptual importance in synthesis studies) include:

(a) source spectra

(b) fundamental frequency

(c) nasal pole-zero pair

(d) transfer function zeros during frication

2. Other Phonological Research

In addition to the developments described above, longer range phonological research has been going on to prepare for handling more complex phonological effects. This work is part of a close collaboration with other ARPA SUR sites which has resulted in three workshops and one group paper [34].

Research on phonology has identified three types of change that affect the sounds of speech. These are segment deletions, segment alterations (both within a word and between words), and segment additions. We include as a special category of deletions those elements which are present in the sound stream, but which may be either missed or improperly identified by an acoustic front end. The details of this last set, of course, reflect the capabilities of the front end and are not constant. We also include as a special category of alterations the segments peculiar to a dialectal pronunciation of a word.

Five factors have been isolated which condition the three types of phonological alteration. The first is dialect. This consists not only of sounds peculiar to a given dialect, but also to the results of invoking specific phonological rules under conditions that are peculiar to a dialect. Thus, some dialects may have a rounded /r/ in such words as "write" as opposed to the plain /r/ of most speakers. Other dialects may devoice vowels under relatively slow speech, but most dialects, if they devoice vowels at all, do so only during rapid speech. Secondly, there are idiolectic variations but the extent of their effect on phonology has not yet been fully determined. Some idiolectic material has already been determined, much remains to be discovered, and a good deal may be found to be dialectal upon future study. This idiolectic material is distinct from the idiosyncratic formant characteristics of an individual's vocal tract. For example, some individuals tend to devoice sonorants more so than others. Thirdly, speed (deliberate, careful, fast, and rapid) has been characterized by the addition and ordering of various phonological rules.

Fourth, style plays some role in conditioning phonology. Speaking style has been restricted in automatic speech understanding research to a nonread, casual, but careful delivery, at least ideally. But in fact, utterances are usually read, and some concessions have been made to this fact, as well as to the simple factor of human inconstancy. Finally, intonation affects segments. The features of pitch, loudness, and length affect segments in the course of expressing emotive and syntactic information.

Two types of dictionaries have been compiled. The first captures a small fraction of the segmental alterations and additions, but a large number of deletions. This dictionary has been used in the November 1973 lexical retrieval component described above. The phonological information encoded therein has been limited only by the system consideration that this dictionary must interface with a front end capable of only limited discernment. Therefore what the front-end cannot see, the dictionary has not bothered to characterize. In the future however, we expect the capabilities of the acoustic/phonetic analyzer to improve and the dictionary will be modified accordingly. A second dictionary has been compiled which marks syllable boundaries. This allows us to encode segmental alterations which reflect differences between certain types of syllable-initial and syllable-final segments. These differences are not phonological, since they are persistent and not a matter of differences between forms. Thus a syllable-final /r/ is always darker than a syllable-initial /r/; this reflects phonetic aspects of English syllable structure, not of dialect, speaking rate, etc. This dictionary is designed to interface with the verification subsystem described above, and the amount of phonological material it reflects is limited to requirements of verification.

Finally, a set of 78 rules has been assembled and issued as a SUR Note [11]. Each rule has a uniform format, explanatory notes, examples, a domain of applicability (within a word or between words), remarks pertaining to matters of intonation, speech rate, idiolect and dialect, ordering specifications with regard to other rules, and comments on any odd or unusual aspect of the rule. Some of these rules are reflected as dictionary entries or the analytic phonological rules described earlier. Most, however, will be implemented in the near future, together with phonotactic information from the verification component, to produce a detailed phonetic dictionary.

IV. DISCOURSE DOMAIN

A. Introduction

This section discusses issues relating to the problem domain in which we are studying automatic speech understanding at BBN. These include reasons for wanting a problem domain, the implications of having one, and the development and characteristics of the problem domains we have used, or currently are using, in SPEECHLIS.

B. Why One Domain?

Two facts justify our desire to limit and characterize a discourse domain in which to attempt speech understanding: 1) the amount of information necessary for the task is incredibly large, and 2) our knowledge of control mechanisms and organizational structures for efficient execution of the task is relatively meager. As a result, any reduction in the amount of information that has to be known to the system brings the problem that much closer to being manageable.

The first implication of limiting the discourse domain is that we can constrain the vocabulary that is needed for conversing intelligently and naturally with the system. This limits the set of words that can be used to compose an utterance, and, from the analytic direction, limits the possible words that can lie behind some region of the speech signal.

Secondly, it enables us to constrain the meaningful use of that vocabulary by characterizing the content of the domain. As a result, one can describe which co-occurrences of words are likely or reasonable to occur and which ones not. It is not enough merely to limit the vocabulary in order to achieve this end. For example, a vocabulary containing just the words (John, California, Lyn, trip, take, need, money), their inflected forms and function words, such as prepositions, determiners, auxiliaries, quantifiers and conjunctions, permits all the following utterances:

(a) How much money does John need for his trip to California?

(b) John tripped Lyn and took her money.

(c) John took up with Lyn in California.

By limiting the content of the discourse domain to travel management, crime stories or even scandal-mongering, one also limits the context in which each word can meaningfully occur. Otherwise, almost any combination is possible by setting an appropriate context. As one poet has shown, even "colorless green ideas sleep furiously" is meaningful, given the right context.

A third result of limiting the discourse domain is that it enables us to characterize how one utterance is likely to follow another, by being able to describe how speakers will use the domain. As a result, one can evaluate the appropriateness of any utterance to its context. For example, by choosing a domain in which certain problems can be solved, one can try to characterize a user's likely problem-solving behavior in that area and its reflection in his linguistic behavior. One could not do this realistically for unconstrained speech.

A fourth result of choosing a specific domain is that it allows one to build a useful, practical system. This in turn encourages people to interact with it. By limiting the domain and building a system which will facilitate solving real problems, we ourselves benefit by being able to collect actual data with which to gain insight into our first three points, and the user benefits by having his problem solved. Although a practical system will not be realized for the spoken aspects of the BBN system for some time due to the time required for speech analysis, the existence of potential users for the subject domain enables us to collect real data on user behavior with respect to the domain.

C. The Lunar Rocks Domain

1. Description of the Domain

Because of its ready availability and its sophisticated syntax and semantics, we selected the LUNAR system [44] for our initial domain. LUNAR is a natural English question-answering system dealing with chemical analyses of the Apollo 11 moon rocks. The LUNAR system understands and answers such questions as:

(d) What is the average concentration of rubidium in high-alkali rocks?

(e) List potassium/rubidium rations for samples not containing silicon.

(f) How many rocks contain greater than 15% plagioclase?

LUNAR also provides a facility for making natural language requests which result in keyphrase document retrieval on the papers from the first Lunar Science Conference held in Houston in 1971. Thus LUNAR can also understand such requests as:

(g) Which papers deal with olivine twinning?

(h) Give me any reports on solar wind flux.

and answer with a set of documents indexed under the appropriate topic or topics.

LUNAR contains a vocabulary of approximately 3500 words and a grammar for an extensive subset of general English. For the initial speech system, we selected a subset of approximately 250 words from LUNAR's vocabulary and a subgrammar of more restricted English from its grammar. The subset of words was selected in such a way that every concept involved in chemical analysis that could be understood by LUNAR would likewise be understood by SPEECHLIS. The only limitation was the number of ways each concept could be expressed, (e.g. the small vocabulary did not contain the names of all the elements), and the number of topics for document retrieval.

The data bases that are available to the SPEECHLIS version of the lunar rocks world were the same as those available to LUNAR: a table containing over 13,000 chemical analyses of the Apollo 11 moon rocks and an inverted file by keyphrase of the papers written for the First Lunar Science Conference. However, the use of the factual data bases is restricted to question answering. No attempt is made to use their information to feed back into the speech understanding process, as additional evidence confirming or denying some reading of the speech wave. Such a feedback loop is envisioned for the travel budget management domain, however, as will be described later.

2. Difficulties in Using this Domain for Speech

There were many difficulties encountered in our use of the LUNAR task domain in our attempt to understand speech. First, it was difficult for us to gain access to informants concerned with problems in lunar geology. Thus, the tasks of building a user model, discourse model, and problem-solving model for this domain threatened to involve an enormous effort which would be completely off the track from the problems of speech understanding, and we decided not to undertake it.

Secondly, from a phonological point of view, there were too many "strange" and unfamiliar words in the lunar geology vocabulary. It was very difficult for non-geologists to formulate or look at sentences containing one or more of these words and utter them in a natural way.

Thirdly, from syntactic and semantic points of view, our own lack of intuitions about how such a system would be used made it very difficult to predict how a user would talk to it or to put in heuristics to evaluate the syntactic and semantic appropriateness of each possible reading of a possible utterance.

Fourthly, because lunar geology is not easily comprehended by a lay audience, demonstrations of the system's capabilities could not easily make a strong impression. The audience rarely knew what a reasonable question was, and cared even less about its answer.

For these reasons, we chose to develop a second domain of discourse. On the one hand, we could study it in parallel with the lunar geology domain to notice domain-specific speech problems, and on the other, we could extend it with the user, discourse and problem solving models that the lunar world lacked.

D. The Travel Budget Management Domain

After considering several possible problem areas in which to develop a new discourse domain for SPEECHLIS, (e.g. inventory control, project management and accounting), we decided upon the area of travel budget management. In this domain, one would expect a system to understand and respond to such utterances as:

(i) What trips did we have budgeted for the speech project as of September, 1973?

(j) Which of them have already been taken?

(k) Give me a list of the remaining trips with the estimated costs.

(l) Nine people will be going to Pittsburgh in April for the IEEE conference.

(m) The registration for that meeting is $40.

(n) If we only send 3 people to London and 1 to Stockholm, will we then be within the budget?

That is, the user will be able to query the data base, add to it, and make both hypothetical and permanent changes to it.

1. Reasons for Selecting this Domain

There were several reasons for choosing this domain, all of which answered shortcomings in the initial domain of lunar rocks. First, within BBN, everyone is to some degree concerned with travel budgets and their management. Therefore, there will be ample opportunity to find informants who will help us in building user and discourse models and will use the system once it is in operation. (Until the new system is completely implemented, we are using the technique of incremental simulation [50] to gather user-system dialogues to guide us in building these models.) A related reason is our own desire to have such a system as a practical tool.

Secondly, except for some place names, the words involved in travel budget management are basically common ones, enabling utterances to be spoken naturally. (Unfortunately, it seems that there is a much larger documented variation in the pronunciation of common words than there is in that of uncommon ones. This has led to at least a doubling in the number of phonemic spellings possible for the same number of words, and has encouraged us to seek an alternative organization for our phonemic dictionary. There is also the potential problem of new words being used to name new places that the system does not know about or to title upcoming meetings. We have decided to finesse this problem by requiring that all new words be entered via the text-based version of our proposed system.)

Thirdly, from syntactic and semantic points of view, the new domain affords many interesting problems that were not likely to appear in the lunar geology domain, such as the problem of hypothetical questions (e.g. sentence (n), above) and ones involving time referents (e.g. sentences (i) - (k)).

Thus far, we have constructed a small vocabulary of about 350 words for the travel budget domain, complete with phonemic spellings and syntactic features, and we are in the process of building a semantic network to represent their meanings and likely contexts. We have also designed a data base and retrieval language for the system, all of which will be discussed in the following sections.

2. Delineating the Domain

The mere selection of the area called "travel budget management" as our new discourse domain was not sufficient to delimit a precise subject area from those which might be termed "related", or to identify the concepts involved in the area and a set of words necessary to speak about it naturally. In this section we describe how we have gone about characterizing the subject matter and use of the domain, collecting a vocabulary for it, identifying grammatically the kinds of sentences most natural to it, and building a semantic representation of the concepts it involves. In this, we have tried as much as possible to formalize the process of delineating a new domain, or at least identify some set of rules and conventions for going about it, so that it will be a cleaner task to do so for other domains in the future.

Our first step was to tell people we were building a travel budget management system and elicit from them a list of questions that they would ask such a system, were it available. In several cases, we actually carried on system simulations, using a person with access to information about our travel budget (e.g. information about trips already taken with regard to expenses, places visited, etc.; information about projected trips; information about upcoming conferences; rough approximations about flight costs, etc.) to simulate the system's projected response to different types of questions. (This also gave us samples of dialogues, allowing us to look at such dialogue techniques as deixis, anaphora and ellipsis. The resulting set of sentences was screened to eliminate those we felt the system shouldn't be able to handle (e.g. policy questions like "Whom should we send to Monterey next spring?", "Which is the least essential trip we have planned?") and those we felt were not in that fuzzy area we wanted to call travel budget management (e.g. requests for travel arrangements like "Is there a flight to L.A. which stops in Salt Lake City?"). This corpus of sentences, 128 in all (see Appendix), has been used for several purposes, one of which was to isolate a vocabulary for the domain. This vocabulary was then reviewed to see if other requests we felt the system should be able to understand were expressible using it. If not, the vocabulary was augmented. This resulted in a vocabulary of about 350 words. Thus the task of describing the domain was accomplished in several cycles: we started with a vague notion of travel budget management in order to elicit specific example sentences from people. These were then used to sharpen the description, to say what travel budget management was and wasn't. This description was in turn filled out with closely related matters which were not touched upon in the necessarily limited set of initial sentences. (A listing of the content words in the resulting lexicon for travel budget management appears in Figure 1.)

a. Syntactic Character

The initial corpus of sentences was also reviewed in order to evolve a characterization of the grammatical forms of utterances most natural to the domain, and the results are presented below. The information gained from this analysis will be used to aid the syntactic component in forming likely hypotheses about the structure of input sentences.

Of the 128 sentences, 98 were questions, 24 were imperatives, and only six were declaratives. Five of the six declarative sentences were in effect commands to enter data into the travel network ("The final cost of the trip was $56.66") and would need to be treated as imperatives. The sixth was in effect a question ("I want to know what trips Bill will take this winter") and would need to be treated as such.

About one third of the questions began "how many" or "how much". Although "how many" was always followed by a noun, usually "people" or "trips", "how much" constructions were most often elliptical. (Of 22 sentences, one was "how much time", one was "how much of the .. funds", four were "how much money" and 16 were "how much" with money implied.) Only two sentences had a prepositional phrase following a quantifier ("Which of those trips have already been taken").

```
(ADJECTIVES (ACOUSTICAL AVAILABLE BIG COMPUTATIONAL CURRENT EACH
     ENOUGH  EXPENSIVE FINAL FISCAL INTERNATIONAL LEFT LONG MANY
     MISCELLANEOUS OTHER PERDIEM RECENT UNANTICIPATED UNBUDGETED
     UNSPENT UNTAKEN UPCOMING VARIOUS))
(ADVERBS (ALREADY ALSO EITHER ENOUGH HOW LONG MORE MUCH NORMALLY
     NOW ONLY PLEASE SO THEN THERE TOO USUALLY YES))
(INTEGERS (EIGHT EIGHTEEN ELEVEN FIFTEEN FIFTY FIVE  FORTY  FOUR
     FOURTEEN  NINE  NINETEEN OH ONE SEVEN SEVENTEEN SEVENTY SIX
     SIXTEEN SIXTY TEN THIRTEEN THIRTY THREE TWELVE TWENTY TWO))
(NOUNS  (ACCOUNT  ACOUSTICS  AIR  AIRPLANE  AMOUNT  ASSOCIATION
     ASSUMPTION  AUTHOR  AVERAGE  BEGINNING BREAKDOWN BUDGET CAR
     CHANGE CITY COAST CONFERENCE CONTRACT COST COUNTRY DATE DAY
     DEAL  DEFICIT DIVISION END ESTIMATE-N EXPENSE FALL FARE FEE
     FIGURE FUNDS GROUP HALF HALVES JOB  LINGUISTICS  LIST  MEAN
     MEETING  MEMBER  MONEY MONTH MUCH NEED NOTE NUMBER OVERHEAD
     PARTICIPANT PEOPLE PERCENT PERDIEM PERSON  PHONOLOGY  PLACE
     PLAN  PLANE  PROJECT-N  PURPOSE  QUARTER RANGE REGISTRATION
     REMAINDER REST ROUND@TRIP SCHEDULE SITE SOCIETY SOME SPEECH
     SPRING  STATUS  SUMMER  SUPPOSITION  SURPLUS THANK@YOU TIME
     TOTAL TRAVEL TRIP VISIT WEEK WEST WINTER WORKSHOP YEAR))
(ORDINALS (EIGHTEENTH  EIGHTH  ELEVENTH  FIFTEENTH  FIFTH  FIRST
     FOURTEENTH  FOURTH  LAST  NEXT  NINETEENTH  NINTH  SECOND
     SEVENTEENTH SEVENTH SIXTEENTH SIXTH TENTH THIRD  THIRTEENTH
     THIRTIETH TWELFTH TWENTIETH))
(POSSESSIVES (HER HIS MY OUR THEIR WHOSE))
(PROPERNOUNS (ACL AI AMHERST APRIL ARPA ASA  AUGUST  BATES  BERT
     BILL  BONNIE  BOSTON  CALIFORNIA  CARNEGIE COLARUSSO COSELL
     CRAIG DAVE DECEMBER DENNIS ENGLAND FEBRUARY ICCL IEEE  IFIP
     IJCAI JACK JANUARY JERRY JOHN JULY JUNE KLATT KLOVSTAD L.A.
     LINDA LONDON LOS@ANGELES LYNN MAKHOUL  MARCH  MASSACHUSETTS
     NASH-WEBBER   NEW@YORK   NOVEMBER   OCTOBER   PAJARRO@DUNES
     PENNSYLVANIA PITTSBURGH RICH RICHARD SANTA@BARBARA SCHWARTZ
     SDC  SEPTEMBER  STOCKHOLM  SUR  SUTHERLAND  SWEDEN  TBILISI
     WASHINGTON WISCONSIN WOLF WOODS))
(PRONOUNS (ANYONE EVERYONE HE HER HIM I IT ME  ONE  SHE  SOMEONE
     THAT THEM THESE THEY THIS THOSE US WE WHAT WHO WHOM YOU))
(SPECIALS (DOLLAR HUNDRED K NO OK THAN THANK@YOU THOUSAND YES))
(VERBS (ADD AFFORD  ALLOW  ANTICIPATE  ARE  ARRANGE  ASK  ASSUME
     ATTEND  AUTHOR  AVERAGE BE BEEN BEGAN BEGIN BEGINNING BEGUN
     BEING  BUDGET  CAN  CANCEL  CHANGE  CHARGE  COMMIT  COMPARE
     CONTINUE  COST COSTING COSTS COULD DEAL DEALING DEALS DEALT
     DID DO DOES DONE END ESTIMATE-V EXPECT FIGURE FIND  FINDING
     FINDS  FOUND  GET  GETS  GETTING GIVE GIVEN GIVES GIVING GO
     GOES GOING GONE GOT GOTTEN HAD HAS HAVE HAVING IS KNEW KNOW
     KNOWING  KNOWN  KNOWS  LAST  LEAVE LEAVES LEAVING LEFT LIST
     MADE MAKE MAKES MAKING MEAN NEED NOTE NUMBER PAY PLAN PRINT
     PROJECT-V  PROPOSE  PUT  RANGE  REMAIN REVISE SCHEDULE SEND
     SENDING  SENDS  SENT  SPEND  SPENDING  SPENDS  SPENT  START
     SUPPOSE  TAKE  TAKEN  TAKES  TAKING TOOK TOTAL TRAVEL VISIT
     WANT WAS WENT WERE WILL WOULD))
```

Figure 1.

Most numbers which occurred were used as quantifiers, usually with "people", and sometimes with ellipsis ("Forget the three people for Santa Barbara and make it just two again"). However, numbers also occur as head nouns ("What's this charge of $350 to 11510") and in number unit pairs ("Add a $30 surcharge for visa costs to the IJCAI", "How many three day trips to California can we afford").

Another third of the questions began with "what". In most of these, "what" was used as a question-word followed by a copula ("What was the average cost"), but in a few "what" was used as a question-determiner ("What job number is being charged for each participant"). The remaining third were mainly yes/no questions with a few beginning with "who", "where", "when", and "why".

Eleven relative clauses occurred, five marked with "that" ("Who are the participants from BBN that plan to attend"), one marked with "which" ("Will the amount of money left in our travel budget cover the trips which have been proposed"), and six unmarked ones ("What is the actual charge of all the trips we have taken"). There were no cases of relative clauses having further relatives embedded within them, a fact of likelihood the grammar can take into account when making hypotheses.

Seven sentential complements occurred, all involving "to". Four of these had the meaning "in order to" ("How much would it cost to send someone to California for a week"), while three did not ("Is John scheduled to go to Carnegie"). In this domain, the fact that a verb can take a "to" complement does not predict strongly that it will. No examples of "for" or "that" complements appeared in the corpus, so these arcs of the grammar will be assigned very low probability of occurrence.

Only two sentences used superlatives ("Which conference is the most expensive?"), and there were no examples of comparatives. Though the present grammar will handle simple superlatives and comparatives, it appears that neither is likely to occur very frequently.

Syntactic structures found in the corpus which cannot be handled at present include possessives, conjunctions, and if...then constructions.

Eight sentences used possessives. Six of them were attached to the first or last names of people ("Cancel Rich's trip to Monterey for June"), while only two of them were not ("What's the state of this year's travel budget right now"). Although possessives present pr blems in speech because they are difficult to distinguish from plurals, we feel this may be a place where we can take advantage of prosodic cues to determine their presence and their scope.

Nine sentences used conjunctions. Four of these sentences used a conjunction to unite two complete sentences ("Change the number of Pittsburgh trips to eight and add Craig to the list of people going"). This situation can be handled quite reasonably by requesting that structures of this sort be offered as two complete sentences or by making a simple addition to the top level grammar which has the same effect. More complicated problems were introduced by the two sentences in which a conjunction followed a long list of items ("What would be the total budgeted amount for four people to New York, four to ACL, two to London, one to Stockholm, plus the other untaken budgeted trips to other places"), and where ellipses occurred either before or after the conjunction ("How much time was there between the London and Stockholm conferences"). These sentences resist rewording in any natural fashion and will be difficult to deal with, not only because of the ellipsis but also because the scope of the conjunction will be hard to determine.

Two sentences employed "if...then" constructions ("If we send five people to California for a week, how many can we send to the IJCAI"). Because they would be very difficult to express in another fashion, we will be expanding the grammar to handle them.

From all the sentences, the open-ended nature of the necessary set of proper nouns was apparent. There will always be the need to enter the names of new places, people, institutions, and conferences, and some method must be devised for letting the user do so in the course of a regular session. Since it will be difficult for the system to recognize that it has heard a new word rather than a sloppy pronunciation of one it already knows, new words will probably have to be entered via text input.

At present we are unable to handle sequences of proper nouns ("John Makhoul", "St. Louis, Missouri") or dates in any form, though we feel it is important to do so. It will be necessary to write a special purpose network for dates, (similar to the special purpose networks for money and numbers already implemented), which will be capable of coping with "July 1st", "July 1, 1974", "1 July", etc. While only three actual dates occurred in the corpus ("September, 1973", "1 July", "April 10th"), there were altogether 24 date expressions (e.g. "this past April", "to date", "right now", "late November", "fiscal 75", "in October"), making clear that the ability to handle such expressions will be a needed one.

The conclusions we have reached here about the likely form of input into the travel budget management system are only tentative: our corpus was drawn from written sentences, and except for two cases of simulating the system, not from a dialogue situation. However, such an analysis is always useful. The scope of the grammar is increased and the likelihood measures we derive can always be altered if we find them faulty.

b. Semantic Character

(1) Major concepts

Not surprisingly, the most important and frequently evidenced concepts appearing in our corpus of sentences on travel budget management were those of <u>budget</u> (in both its noun and verb senses) and <u>trip</u>. (On word count alone, the only word appearing more often than "trip" or "trips" in the corpus was the word "the".)

Just to say that these are the most important concepts in the domain is not enough: we must look at how far we are allowing these concepts to be broken down and in what directions, in order to characterize what people can and will be allowed to say about them to the system. For example, although trips can be analyzed down to the clothes packed for a trip, a particular seat on the plane, a room number in a hotel or the names of friends one is staying with, etc., they need not be, in order to speak naturally and freely on travel budget management. The properties of a trip that will concern us in this limited domain are:

(a) its cost, both estimated and actual, broken down by travel fare, accommodation, food, and miscellaneous

(b) its destination or set of destinations

(c) the person taking the trip

(d) its trip number (an internal BBN convention)

(e) its duration and when it was/will-be taken

(f) the account number being billed against

(g) the budget item it is an instance of

(h) its status - whether it is merely planned or has been taken, whether it is an instance of some budget item or may be termed "unbudgeted".

We will not be concerned with particular flights taken, the names or locations of hotels stayed at, or daily activity schedules for the person taking the trip. Because "trips" are understood to the level of detail given above, they become objects which:

(a) can be added to or cancelled from the budget, planned, proposed or budgeted for;

(b) can be taken by a person to various places for some length of time at some point in the year;

(c) cost some amount of money or have money spent on them;

(d) can be afforded (or not);

(e) can be queried with respect to any of the above properties.

The second important concept, "budget", is understood in its noun sense both as a plan for spending money and as a record of how much has been spent and on what. Specifically, we know it as something which:

(a) is associated with a given contract (or equivalently, a given account)

(b) may be recomputed several times during the year, but only one of these will be "current" at any one time;

(c) is a list of "budget items" (i.e. trip descriptions), whose minimal content is the number of people-days to place X (e.g. three five-day California trips) and an approximate or actual cost;

(d) contains a certain amount of money which is allocated either partially or completely to the budget items.

As a result, it is reasonable to add, delete and change items in the budget; to ask how two budgets differ; to ask how much money is left in the budget; to make hypothetical budget changes to observe their consequences before possibly making a new budget; and to ask to see its current state.

In its verb sense, "budget" is understood to mean the act of adding a new entry to a budget. Since we have established that entries are trip descriptions and budgets are made by people, the verb "budget" will occur in such contexts as budgeting trips to some place, budgeting money for some trip, and budgeting people to go on some trip.

The remainder of the concepts which compose the domain of travel budget management are ones which allow "trip" and "budget" to be thought and talked about in the above terms. These concepts include those of places, money, dates, conferences, people, and means of transportation.

(2) Building a Semantic Network

In the process of building a new semantic network to represent the objects and concepts involved in travel budget management, their interrelations, and the ways they may be discussed, we have also been trying to introspect and characterize procedures we use in the process. Ideally, we would like to remove as much of an ad hoc nature as possible from the process and have a system in which a network could be built up through English interactions. Failing this (since it is a non-trivial open problem in the field), we would at least like to develop some convenient set of rules and conventions with appropriate supporting mechanisms to enable a researcher to rapidly construct a semantic network for a given new domain or expand a given one.

At the current time, the semantic network for travel budget management has not yet been completed, nor have we gotten a total feeling for a characterization of the building and enlarging process. For completeness though, we present below an annotated partial example of the procedure we are currently using for entering new adjectives into the semantic network. (If a word has more than one syntactic part of speech, a separate procedure will be followed for each one. Note that since the semantic network is being used to predict the contexts in which each content word in the lexicon can occur, syntactic information such as part of speech, and the ability to take various sentential complements has semantic import as well.) In this exchange, we envision the system asking the questions and the network builder providing the answers about the word and how it is used. Currently, the network builder just answers a written set of questions and is free to enlarge this list as seems useful for capturing correctly the appropriate uses of a word. (The system's intended part in the dialogue is underlined.)

```
>Enter new word: big
> What is/are its part(s) of speech? adj
> Does it form comparatives and superlatives? yes[*]
> What can "big" modify? objects and aggregates
>Can you give me one or more examples of "objects" in the
domain?

     1. account          2. budget

>Can you give me one or more examples of "aggregates".

     1. division         2. group          3. project-n

>"group", "division", and "one sense of "project-n" belong
to the class "groups of individuals". Are "groups of
individuals" "aggregates".

     yes
```

3. A Factual Data Base for Travel Budget Management

a. Description

From our simulations and discussions of a Travel Budget Management system, it appears that the factual data base for the domain will have to contain several different groups of facts. These include budget items (specific commitments of funds, vague plans, and options), information on specific trips (either taken or planned), costs of traveling between cities, geographic

[*]The ability to form comparatives and superlatives implies that the adjective is describing some scalable property of an object and hence will also occur in such constructions as "how X is", "too X", "very X", "X enough".

information, and facts about meetings (conferences, symposia, etc.). The facts are heterogeneous and subject to several different operations. Users will refer to them, inquire about their properties, modify or make conjectures about them.

We have chosen to represent this factual data in a semantic network parallel to that used by the Semantics component of SPEECHLIS. (This network resembles that used by Shapiro [41] in the MIND system. We are taking advantage of an existing implementation due to R.M. Kaplan and extended by R. Burton and B.L. Nash-Webber.) Several factors motivated the choice of this representat ..n.

First, the system needs to represent diverse facts in a flexible manner, allowing information to be given at different levels of detail. Secondly, the Semantic component of SPEECHLIS needs to make use of the factual data base. This would be useful, for instance, if semantics has a theory which concerns a specific trip. If it can find a referent for that trip in the data base, it will have more confidence in the theory. Thirdly, a semantic network facilitates many types of inferencing which are useful in information retrieval to avoid storing all possible relations between data items explicitly. For instance, one might request a list of all West Coast trips. A correct response to this request would include trips to SRI, Santa Monica, the 3rd IJCAI, etc. In order to retrieve such trips, one must somehow associate these destinations with the general description "West Coast trips". One way is to store this information directly. Another is to infer from the facts that

(1) California is a West Coast state;
(2) Palo Alto is in California;
(3) SRI is in Palo Alto.

that a trip to SRI is a West Coast trip. This type of inferencing is very convenient in a semantic network. Finally, it allows objects to be referred to in many different ways. For instance, a trip may be described by the person who took the trip, its date, destination, or any combination of these descriptions.

The following advantages of semantic networks help meet these goals.

(1) The structure is consistent with the network used by the semantic component. This consistency will enable semantics to access the factual data base easily.

(2) The two way links in the network provide retrieval keys for all types of facts. For example, one may retrieve all trips taken to some location or alternatively, all locations visited on some trip. This simplifies the retrieval task.

(3) Recent research [10,12], has shown that semantic networks are a useful representation in which to consider plausible inferences of the type done by people every day. We expect to need such a capability in responding to requests in the travel budget domain.

(4) Much software for building and searching semantic networks already exists.

The following Figure illustrates a piece of the network for representing a typical trip:

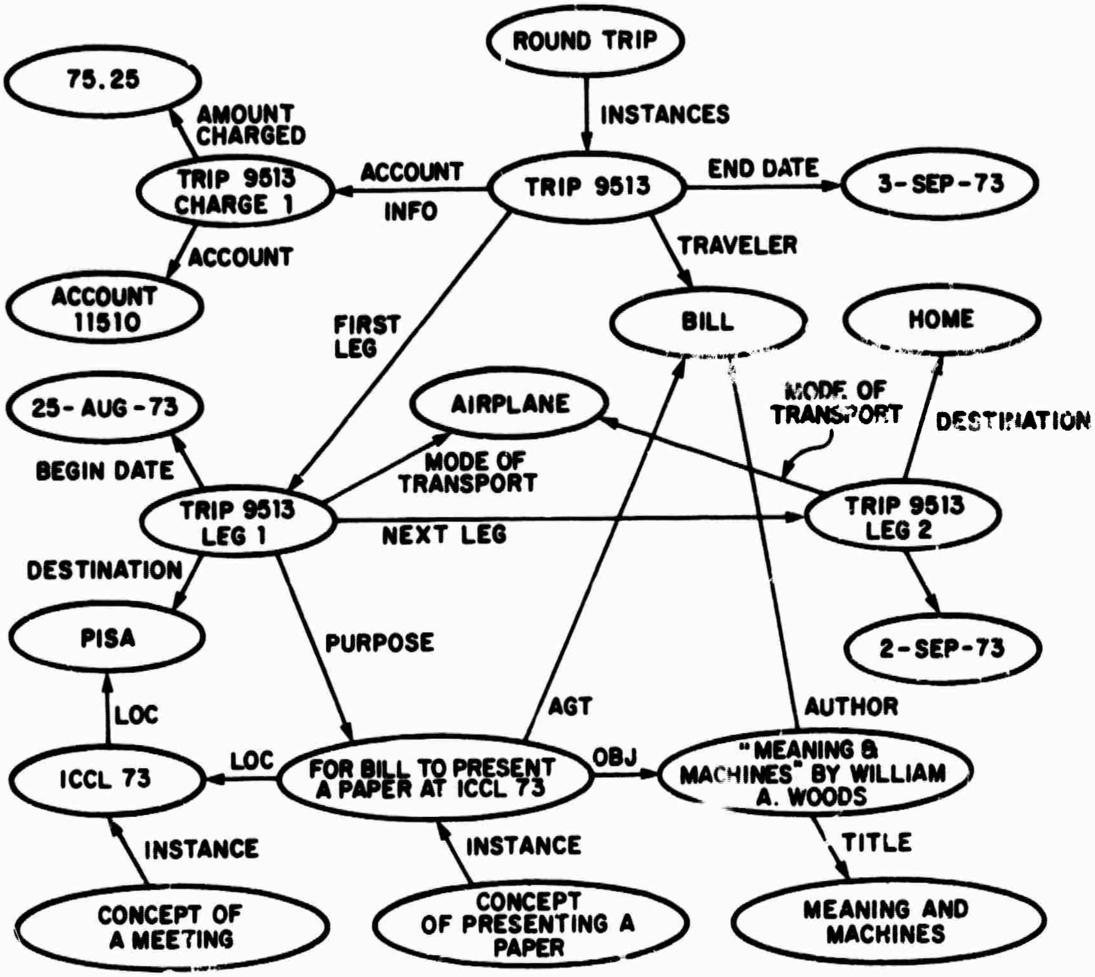


Figure 2. Travel Budget Management Data Base (an excerpt)

b. Construction and Retrieval Functions

As noted above, low level routines for building and searching semantic networks currently exist. In order to further simplify the process of constructing the data base, higher level programs have been written that reduce a large part of this effort to a clerical task. The function DLGTRIP can prompt a user for the basic facts about a trip and then build the semantic network representation for that trip. A sample protocol is shown in Figure 3. Similarly, the function BUILD-FARE simplifies the process of building a representation for the cost of traveling between two cities.

```
↑↑(DLGTRIP T]
 TRIP NO. :9513
 ACCT AND AMOUNT - PAIRS :((11510 75.25]
 TAKEN BY :BILL
 NUMBER OF LEGS :2
 LEG 1
      BEGIN DATE :25-AUG-73
      PURPOSE :(FOR BILL TO PRESENT A PAPER AT ICCL 73)
      DESTINATION :PISA
      MODE OF TRANSPORT :AIRPLANE
 LEG 2
      BEGIN DATE :2-SEP-73
      DESTINATION :(HOME)
      MODE OF TRANSPORT :AIRPLANE
      END DATE :3-SEP-73
```

Figure 3
Sample protocol for building semantic network
for a trip (computer printout is underlined)

In retrieving information from a semantic network it is necessary to find all nodes related by a relation, R, to a given set of nodes, T. The query language for stating retrieval requests is implemented via the function BOOLFINDQ whose arguments describe the set of nodes to be retrieved. Each argument takes either of the following forms:

(1) (R, T) where R is a defined relation and T specifies a node or set of nodes.

(2) an arbitrary LISP expression that evaluates to an ordered list of nodes.

To aid in performing typical retrieval operations, four functions are provided (to be used within the query language). BF-OR and BF-AND take arguments as BOOLFINDQ does, and respectively return the union or intersection of the sets of nodes described by its arguments. BF-SDIFF takes two arguments of the form given above and returns the set difference of the nodes specified by its first and second arguments. PRED-CHECK takes three arguments:

a node or a node list
a property (i.e. a link without an inverse)
a predicate.

PRED-CHECK first retrieves the value for the given property for each node in the node list. It then returns the subset of nodes for which the predicate, applied to the corresponding property values, evaluates to a non-NIL value. For example:

```
(PRED-CHECK
   (BOOLFINDQ (TRAVELER (QUOTE (JOHN BILL))))
   END/DATE
   (FUNCTION (LAMBDA (DATE)
                (EARLIER-THAN DATE 31-DEC-73))))
```

Will retrieve all trips taken by either JOHN or BILL that were completed prior to December 31, 1973.

In addition to domain-independent retrieval functions like BOOLFINDQ, there are also special purpose retrieval functions for trip and budget information. One example is the function FARE. It will determine the fare from city A to city B via a given vehicle (which defaults to airplane). This would be used to answer questions such as:

"What is the cost of traveling from Boston to Los Angeles?"

Other examples of specialized retrieval routines include TRIP (for retrieving all trips specified by a set of descriptors), TRIPLEG, PURPOSE, and DESTINATION. These procedures will construct and execute instructions in the formal query language.

Several objectives remain to be attained; including constructing a significant data base, specifying a formal query notation and writing further specialized retrieval functions.

4. Multi-Level Use

We have designed the travel budget management system in such a way that it will not be constrained to spoken input. It will be able to accept input via three separate channels: natural language speech, natural language text, and text in a formal retrieval language. There are several advantages to having this ability. Being able to use the formal retrieval language directly will provide an efficient, practical way of managing travel budgets, a facility we can use within the project. It also gives us a convenient way of entering the names of new places and descriptions of meetings, a difficult process in text and an impossible one in speech, given current knowledge. The natural language text system will provide a ruler against which we will be able to measure the system's syntactic and semantic performance: we will be able to see what the system can parse and interpret without the additional problems caused by speech. It will also provide the criterion of correctness against which to measure the performance of the speech system.

5. Extending the Lexicon

In keeping with the goals set out in the Final Report of the Study Group on Speech Understanding [33], we have also been considering non-trivial ways of extending our initial vocabulary of 350 words to one of 1000 words. (A trivial way would involve, for instance, adding 650 new place names.) The way we have chosen is to choose a topic area related to travel budget management and extend the range of concepts (and hence words) admissible in the system. Several areas related to travel budget management were suggested by our initial corpus of sentences (e.g. managing other types of resources besides travel funds, keeping track of people's schedules and movements, and arranging or helping to arrange trips). One of these will probably form the basis of the above extension. Independent of the area chosen, a major consideration we will have in expanding the vocabulary will be to organize the lexicon for maximization of efficient retrieval by taking advantage of phonetic, syntactic and semantic relationships. Work has already begun on re-organizing the small lexicon to take advantage of the syntactic as well as phonetic proximity of the words. For the expanded lexicon, we hope to bring in semantic nearness as well.

V. OVERALL CONTROL STRATEGY

A. Introduction

By means of incremental simulations with various components of the system implemented as a combination of code and people, we have been attempting to evolve effective strategies for the overall process of analyzing and "understanding" speech signals. For the sake of discussion and experimentation, we have thought of this strategy as being embodied in a control component whose task is to decide which of the other components to call and when. It may be that in the final system most of the control component may be distributed over the various other components of the system in little bits of code and conventions, leaving only a vestigial component, or none at all which can be isolated and referred to as the control routine. (Already many of the strategies for trying alternative ways to find a word match in the feature lattice have been incorporated into the lexical retrieval component and no longer have to be considered by a person who simulates the control component.) However, the consideration of this component either as a reality or as a fiction is beneficial in formulating and simulating various overall strategies for the operation of the total speech understanding system.

For the most part, we have been focusing our attention in the control area on the mutual interactions among the control component and the syntactic, semantic, and pragmatics components. Specifically we are working on ways to use the syntactic, semantic, and pragmatic information available to guide the creation, evaluation and growth of alternative theories or hypotheses about the structure and content of the utterance being analyzed. The framework which we have been considering is one in which each such theory is represented as a specific data object which we can create and refine and to which we can attach various evaluation parameters reflecting the status of the theory and the confidence we have in its being correct. In addition, we can associate with a theory various events which may or may not occur somewhere in the analysis of the utterance that would affect the status of the theory in some way. These are awaited by event monitors which essentially watch for such events and cause the associated theories to be reconsidered when they occur.

Event monitors are the functional equivalent of the "demons" used in Carl Hewitt's PLANNER language [18] and similar notions of "active elements" that are sprinkled throughout the artificial intelligence and problem solving literature. They are also like the "interrupts" which make time-sharing systems and other such applications of computers possible. Event monitors can be created to watch for the discovery of a particular word anywhere in the sentence, the stimulation of a concept node in the semantic network, or the discovery of any word beginning or ending at a particular segment boundary in the feature lattice. When such an event occurs, an event notice is constructed from information contained in the event monitor about the associated theory which created the monitor and why the monitor was created. The event notice represents a potential theory which may be formed as a refinement or modification of the original theory.

One of the critical problems that the overall control strategy must solve is how to avoid excessive duplication of effort and the combinatorial explosion of possible theories that would result. It is important not to unconsciously generate the same theory in many different ways. For example, there are usually several different ways to grow the same theory. By checking whether any two event notices would result in the same theory, however, we can avoid this duplication. The major reason that this is an issue is that most existing techniques for eliminating this sort of duplication consist of choosing a particular order in which to combine the pieces and constraining the algorithm to combine pieces only in that order (e.g. left-to-right parsing -- See [49] for a discussion). In the speech environment, the high probability of errors in the signal analysis makes it possible that some crucial piece may be missing. In order to propose or look for it explicitly, it is necessary to first combine the remaining pieces without it. If one could be assured that his order of combining pieces were such that the missing piece was guaranteed to be the last one, then these two objectives would not be inconsistent, but that seems like an impossible condition to obtain.

Our present control strategy embodies a set of procedures which we have found useful as a result of our experience with various incremental simulations of the speech understanding system. Many of the specific details and the justifications for them are given in the chapters describing the syntactic and semantic components. The general outline of the control strategy, however, is as follows: The control strategy first directs the search for words anywhere in the sentence that are longer than two segments and match well. Then the proposals which have accumulated are processed: specified words are matched at specified positions and entered in the word lattice if their match quality is better than average. Words which are likely to appear at the beginning of the sentence are then matched at position 0, and they are entered if their match quality is not poor. After this is done, accumulated proposals are again processed, and then event-notices are processed. After processing the "good" event notices, the next step is to allow Syntax to do what it can with the theories which contain adjacent words, or words with small gaps between them. Eventually, if the analysis is successful, a complete theory will be constructed which covers the input and is syntactically and semantically acceptable. In the current stage of development of the control component, we stop when the first such theory is found. It is possible, however, that one should continue to look for other complete theories with comparable scores before quitting, and then call upon pragmatic considerations to choose between any competing complete interpretations. This and many other details of operation are currently undetermined, and the current structure of our control component is to be considered tentative and subject to continual development.

In the remainder of this chapter we will describe the techniques we have used to integrate many different sources of knowledge into a coordinated speech understanding system. This will include an introduction to the framework of concepts, data objects, queues, and programs which we have used to express strategies for forming and evaluating competing hypotheses about the interpretation of an utterance, a rough description of our current overall strategy, and an example of its performance. Many more details will be given in the chapters on syntax and semantics.

B. Overview of the Control Framework

1. Data Objects

The control framework assumes the existence of programs which have access to various sources of knowledge. For example, acoustic-phonetic and phonolcgical programs operate on a digitized wave form to produce an acoustic transcription of the utterance in the form of a collection of SEGMENT descriptors. By a segment we mean a portion of the utterance which is hypothesized to be a single phoneme. Each segment has a description which could in principle specify the phonetic identity of the segment, but in general merely constrains this identity to one of several phonemes. Alternative hypothesized segments may overlap in the utterance, resulting in a lattice of segment descriptors rather than a single string. Figure 1 gives an example of such a SEGMENT LATTICE. This structure allows for the representation of uncertainty or ambiguity both in the identity of a segment and in the determination of the segment boundaries.

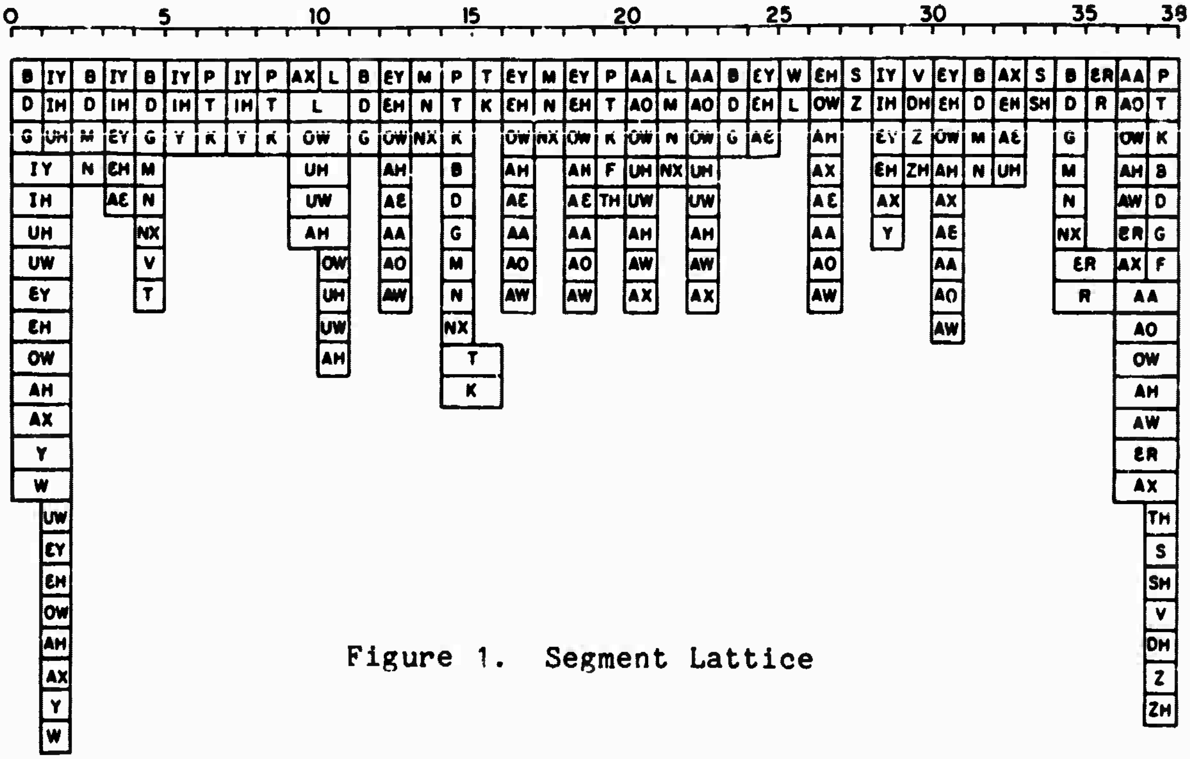
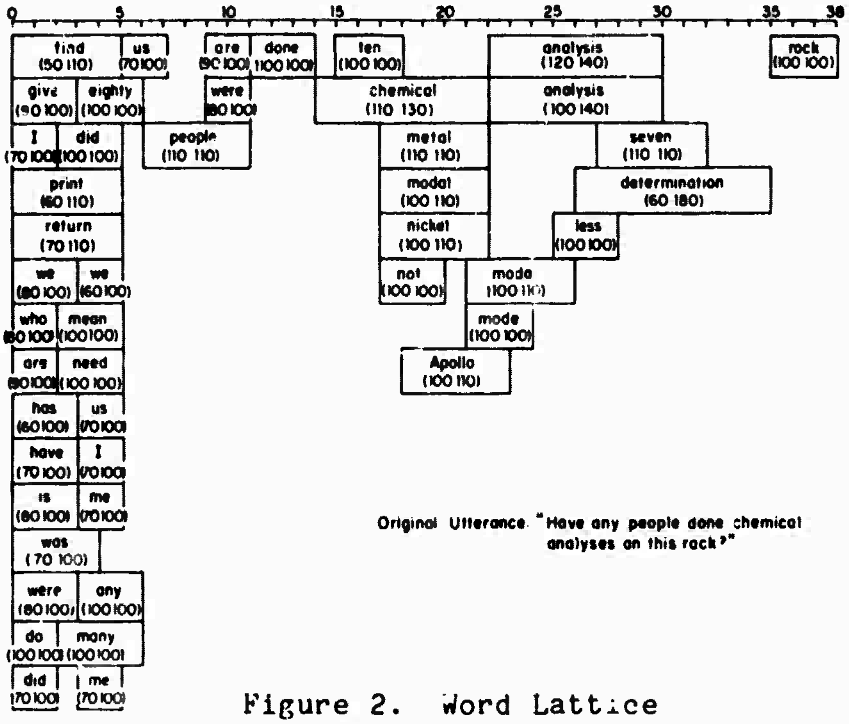


Figure 1. Segment Lattice



Original Utterance "Have any people done chemical analyses on this rock?"

Figure 2. Word Lattice

Lexical retrieval and word matching programs are available to map sequences of segment descriptions into words. They do this by matching PHONETIC SPELLINGS of the words in the vocabulary against sequences of adjacent segments. The correspondence between a single phonetic spelling of a word and a segment sequence is called a WORD MATCH. Since the acoustic transcription may make errors in the detection of segments, word matches involving missing or extra segments may also be made. The quality of the match is one indication of the likelihood that the word actually appears at that place in the utterance. Word matches to be examined by Syntax, Semantics and Pragmatics programs are entered into a WORD LATTICE. (Such a lattice is illustrated in Figure 2.) In this figure, for example, the word "mean", spelled phonetically [min], or to use our computer representation [M IY N], matches from position 2 to position 5 in the lattice, while the word "any", spelled [ɛni] or [EH N IY], matches from 3 to 6.

Each phoneme in the above two spellings satisfies exactly the phoneme description of its corresponding segment. We do not assume however that the correct phonemic identity of a segment will always be among the set of phonemes postulated by the acoustic-phonetic and phonological programs. Rather we assume that if they err, the correct phoneme will be similar in acoustic characteristics to those given. For example, at the beginning of the segment lattice, the first two phonemes of the word "give", spelled [gIv] or [G IH V], match the segment descriptors perfectly. The third, [v], is sufficiently close to [b] acoustically, that a word match is made for "give" and entered into the word lattice. However, since the acoustic transcription is the best evidence we have of what the utterance was, our confidence in "give" actually beginning the utterance is less than if each of its phonemes had matched perfectly.

Interacting with the word lattice, the higher level components of the system (syntax, semantics and pragmatics) form internal data objects called THEORIES representing hypotheses about the original utterance. A theory contains a non-overlapping collection of word matches which are postulated to be in the utterance, together with syntactic, semantic and pragmatic information about this collection and scores representing the evaluations of that theory by various knowledge sources.

Theories grow and change as additional bits of evidence for or against them are found. A principal mechanism for accomplishing this is the creation of MONITORS. A monitor is a trap set by a hypothesis on new information which, if found, would result in a change or extension of the monitoring hypothesis. However, the reprocessing that is called for when a monitor is noticed is not done immediately. Rather an EVENT is created, pointing to the monitor and the new evidence. This event is evaluated to decide if and when to do it.

The use of EVENTS which are not immediately executed but are placed on a queue for later execution at the discretion of the control component is one of the devices whereby the control component manages competing theories about the utterance and constrains its attention to regions of its search space which are likely to pay off. The control component functions somewhat like a time-sharing system in that it is simultaneously managing a number of relatively independent processes (the different partial theories), devoting resources to each with differing priorities (although unlike a time-sharing system it is not interested in guaranteeing that each of the processes will eventually get done). However, instead of interrupting one of these processes at the end of some arbitrary quantum of resource which has been consumed in order to devote resources to another, it is arranged so that all such processes will perform at most a limited amount of computation before "terminating" in the creation of one or more events which are placed on the event queue for further consideration, with scores to be used to determine the priorities for consideration (or perhaps in monitors which may later create such events). Thus, the refinement and development of a theory proceeds in small steps, (each terminating in the creation of an EVENT) which return continually to the control component for evaluation in comparison with other partial theories to determine which ones should be given further development.

In addition to waiting for new information (by setting monitors), the higher level components can also actively seek out information. One way this is done is by PROPOSALS. A proposal is a request to match a particular word or set of words at some point in the utterance. Any of the higher level components can make proposals.

A short example should illustrate the above concepts more clearly. Notice the robust word match for "chemical" in the word lattice shown in Figure 2. The semantics component knows about CHEMICAL ANALYSES and CHEMICAL ELEMENTS, but not about CHEMICAL as an independent concept. Since "chemical" matches well, semantics might postulate that one of these concepts is being designated. It could propose "analysis", "analyses", "determination"(all naming the first concept) and "element", requesting them to be compared against the segment lattice, right adjacent to "chemical". Since "analyses" and "analysis" match well, events would be created, linking the hypothesis for "chemical" with those for "analysis" and "analyses". Given that CHEMICAL ANALYSIS refers to the amount of each major element in some rock, e.g. "chemical analyses of fine-grained lunar rocks", any hypothesis created for "chemical analyses" will monitor for an instantiation of the concept ROCK. If found, it will give additional support to the theory that what is being discussed is indeed the chemical analyses of some rock.

2. Evaluation Mechanisms

A notion central to the control framework is that of evaluation: one cannot afford to spend time on activities unlikely to produce good results. The various scores associated with a theory are used by Control to allocate its resources to where it expects to achieve results. In this section, we discuss how knowledge is brought to bear in computing these scores.

The score of a word match depends on how well each of the phonemes in the phonetic spelling matches the corresponding sound description in the segment lattice. Among the factors taken into account in making this match are such things as:

a) A priori information about the similarity of sounds (e.g. [i] is more similar to [I] than to [a].)

b) Cues from comparing the actual duration of a segment with duration information derivable from the phonetic spelling using vowel tenseness and stress.

c) The likelihood of missing or extra segments. This is determined both from empirical studies of the segmentation errors which are made by the acoustic-phonetic programs and from phonological rules which indicate the sounds in each phonetic spelling which are likely to be missing or extra.

d) The length of the word. Long words which match well get a boost in score because it is relatively unlikely that good long word matches would be detected at random.

The score of a theory is a weighted sum of its lexical, syntactic, semantic and pragmatic scores. The lexical score depends on the average word match score for the words in that theory, the number of adjacent word matches, and acoustic effects at their boundaries. The semantic score is based on an evaluation of the conceptual structures that semantics has built, reflecting whether they are complete or lack some obligatory component. In the latter case, semantic confidence in the theory is lowered.

The syntactic evaluation is based on the ability to assign syntactic structure to the hypothesis. Using an augmented transition network grammar [45] and a parser capable of working with disjoint sequences of word matches, the syntactic component tries to parse each such sequence and decide whether sequences could be joined into a larger syntactic structure. If a word match sequence fails to parse, or if two nearby sequences cannot be bridged in any way, syntactic confidence in the hypothesis will be low.

Currently, SPEECHLIS contains very limited pragmatic knowledge: only the most rudimentary speaker and context models are available for use in evaluating a theory. Observing the relationships postulated by syntax and semantics, the pragmatic component evaluates the likelihood of an utterance that would contain them. For example, in the context of question-answering, questions and commands are more likely than statements: so pragmatics looks for syntactic evidence of sentence type in making its evaluation. The question-answering context also makes certain semantic concepts more likely than others. For example, the concept of the machine giving the user something or of the user needing something is more likely to be expressed than any particular concept, such as that of spectrographic analysis. The pragmatic component uses the conceptual structures that semantics has built to evaluate their likelihood of occurrence. (This evaluation is user independent in the November 1973 system, but we expect eventually to deal with a dynamically developed model of the user's interest.)

There is a further evaluation based on the consistency of the semantic and syntactic structures. Associated with each conceptual structure that semantics has built is a condensed description of the ways in which that structure might be realized syntactically. If none of the structures that syntax can build correspond to these, this discrepancy lowers the likelihood of the theory actually representing part or all of the original utterance.

An event is evaluated in the same way as a theory: that is, the score of an event will reflect the score of the suggested new theory.

3. The November 1973 Control Strategy

Within the framework of word matches, theories, evaluation mechanisms, etc., a preliminary control strategy was implemented for the November 1973 system. In this strategy, the proposals, theories and events that occur during processing are evaluated and placed on three separate queues, ordered by the scores of their elements. The basic characteristic of this strategy is to select elements from the tops of these queues and process them.

The first activity of the control programs is to call the acoustic-phonetic and phonological programs to construct an initial segment lattice from the speech signal. A word lattice of robust word-matches is then constructed by a program which scans the segment lattice with the aid of the dictionary looking for "good", "big" word matches. In addition, a set of words which are pragmatically likely to begin an utterance are matched at the beginning of the segment lattice. As each such word match is found, it is entered into the word lattice and given to the semantic component for analysis. If the word has semantic content, a theory is created for the word match, designating all semantic contexts in which it could appear. If a monitor is noticed indicating that a word fits into the semantic context of a theory which was created earlier, an event is created which associates the new word match with the old theory. Proposals for specific content words which are likely to appear adjacent to the new word match are created and added to the proposals queue.

For each new word match, appropriate inflectional endings and auxiliary verbs are matched against the segment lattice and associated with the word match if they match well.

After the initial set of robust word matches are examined, the proposals that are likely to be productive are processed, thus introducing new word matches and triggering a new round of semantic analysis. The events at the top of the event queue are then handed back to the semantic component for further processing. For each event, a new theory is created with a modified semantic context and entered into the theory queue. This may result in additional events, as Semantics notices other word matches in the word lattice which fit into the modified context. In this way, Semantics assembles meaningful sets of content words.

As new theories are created, each is examined to determine whether it might be fruitful to call upon syntactic knowledge to develop further support for it. Since the number of possible parsings decreases with the number of adjacent or "close" word matches, this decision is made on the basis of the number of adjacent word matches in the theory, the size of the gaps between word match sequences, and the absence of content words in the word lattice which would be added to the theory by semantics.

Syntactic knowledge is used to postulate grammatical structures that may obtain among the words in a theory. For example, for "...people done chemical analyses...", syntax could suggest that "people" is the subject of the verb "done", "chemical analyses" is the noun-phrase object, and that an auxiliary verb appears somewhere in the utterance (probably at the beginning) to modify the past participle "done". Such grammatical information is checked for consistency with the postulated semantic structures, to determine for example whether it makes semantic sense for "people" to do something. Function words (e.g. determiners and prepositions) which are likely to appear adjacent to a sequence of word matches are proposed by Syntax in the context of these grammatical structures and added to the theory as a refinement if they are found. Each small gap between sequences of word matches is analyzed, and a strong attempt is made to find a small word which fits. If none is found, it is likely that one of the word matches adjacent to the gap is wrong.

C. An Example

To illustrate the operation of the above control strategy, we will consider a specific example. The segment lattice shown in Figure 1 was constructed by hand from a speech spectrogram during a study of human performance in spectrogram reading experiments [21]. The word lattice shown schematically in Figure 2 was constructed from it by the control component by looking for robust word matches and possible adjuncts (inflections and auxiliaries) and by trying to match pragmatically likely words in sentence initial position.

Following the first pass in which word matches were entered in the word lattice and given to Semantics for processing, there were 42 theories and 48 events. (Some pruning was done to eliminate unlikely events.) The five events at the top of the event queue were ones linking "chemical" and "analyses", "modal" and "analyses", "chemical" and "analysis", "modal" and "analysis", and "metal" and "analyses". (One can analyze a rock for its metal content.)

Processing these five events led to the creation of five new theories and 55 new events. At this point, the best events called for linking:

(a) "give" (initial position) and "chemical analyses"

(b) "give" (initial position) and "modal analyses"

(c) "give" (initial position) and "chemical analysis"

(d) "print" (initial position) and "chemical analyses"

(e) "have" (initial position) "done" and "chemical analyses"

Notice that the top four events were quite reasonable though incorrect. Five new theories and 20 new events were created during this round of processing.

The next round of event processing brought the following five events to the top of the queue:

(a) "have ... done chemical analyses" and "people"

(b) "have ... done chemical analyses" and "rock"

(c) "give ... chemical analyses" and "me" (following "give")

(d) "give .. chemical analyses" and "us" (following "give")

(e) "give ... chemical analyses" and "I" (following "give")

Notice that the top two events were each filling up a different semantic role in the concept of doing a chemical analysis - the agent of the doing and the object of the analysis. As to the "give I" event, semantics does not know that this is syntactically incorrect. Again five new theories were created during this round, but these resulted in only the five events shown above.

At the start of the fourth round of event processing, the five best events were:

(a) "have ... people done chemical analyses" and "rock"

(b) "have ... done chemical analyses ... rock" and "people"

(c) "give me ... chemical analyses" and "rock"

(d) "give us ... chemical analyses" and "rock"

(e) "give I ... chemical analyses" and "rock"

Notice that the top two events would result in the same theory. However, before a theory is created, the control strategy checks that no such theory already exists. If one does, processing is halted on that event so that duplication does not occur. (Recall that this ability to arrive at the same theory from several directions is necessary since it allows us to put together incomplete structures, regardless of which pieces are missing.) The four resulting theories were semantically complete: both agent and object of "doing" had been identified, as had the object of "chemical analyses", and agent, recipient and object of "give". At this point, Semantics could not contribute anything to these good theories, and they were sent off to Syntax.

Syntax noticed the determiner "any" in the word lattice which could precede "people" syntactically, and it created an event which would refine the first theory with the word match for "any". In addition, Syntax proposed determiners before "rock", since none occurred in the word lattice. This and additional proposals brought word matches for "this" and "in" into the word lattice. These were added to the theory by Syntax, resulting in a semantically meaningful, grammatically correct one which spanned the utterance. This was, at the time, a sufficient criterion for accepting the theory "Have any people done chemical analyses on this rock" as a correct understanding of the utterance.

D. Conclusion

Both the control framework and strategy presented above are incomplete since many problems have still to be faced. Our most difficult current problem involves recognizing the state when the system is just thrashing around, when no theory deriving from our current strategies is likely to emerge as a good candidate for the whole utterance. We need to use our knowledge sources to decide which pieces of existing theories are most reliable, and which pieces should be tossed out. To get a better feeling for the possibilities, we expect to run additional incremental simulations in which a person simulates the parts of the system which are not yet formulated to gain insight into how they might work and monitors the behavior of the rest.

Another pressing problem is the need for a more rigorous foundation for measuring confidence in evidence and combining such measures into measures of confidence in theories and events. As complexity increases, our current methods will become more difficult to manage. We have made a good start in this direction in the design of the new acoustic/phonetic recognizer and lexical retrieval components and hope to do the same for the rest of the control strategy.

VI. THE SYNTACTIC COMPONENT

A. Introduction

The syntactic component of the speech understanding system serves a dual role. Its primary function is to make a syntactic evaluation of the words in a given theory (i.e. to verify or deny the syntactic well-formedness of the set of words in a given theory). It is also responsible for predicting words which have been missed by the lexical retrieval routines but which are syntactically motivated by words that have already been found and the syntactic structures in which they can occur. Thus it may extend a theory by including additional words from the word lattice, and by proposing new words to be looked for at particular points in the utterance.

Because the syntactic component comprises two major parts, the grammar and the parser, there have been two principal areas of research in natural language syntax as part of the speech project at BBN. One is the development of a grammar for a large subset of spoken English. The other is the development of a parser as part of the speech understanding system.

B. The Grammar Formalism

The augmented transition network formalism was chosen as the representation for our grammar because it 1) allows us to draw on our previous experience with the NASA LUNAR system, and

2) it permits the production of "deep structure" analyses like those produced by a transformational grammar without the impractical combinatorial explosion that results from using reverse transformational rules. Indeed, the transition network model provides not only a more efficient way of producing equivalent types of structures, but also theoretical solutions to a number of problems with the traditional transformational formalism [44,45].

It also furnishes many useful insights into the natural language understanding process [19], though it was not originally conceived of as a psychological model for the types of processing which humans perform in analyzing sentences. In addition, it enables a clear interfacing of the grammar to semantic and pragmatic components of a total natural language understanding system and lends itself readily to investigating the problems of continuous speech understanding.

For a complete description of TNG's and a text parser using them, see [44,45]. Briefly, a TNG looks something like a finite state network, with two important additions. The network may be recursive, that is, the label on some arc may call for a structure created by recursively re-applying the network. Second, there may be a list of ACTIONs on each arc whose purpose is to perform tests or to create bits of tree structure and store them in REGISTERs which may be thought of as free variables whose values are accessible to subsequent arcs. In this manner, register contents can be combined and built up to finally produce a deep structure analysis of the sentence.

Figure 1 shows a diagram of a simple TNG. The names of the states are within the circles. The types of arcs shown are: CAT X, which looks at the string for a word of syntactic category X; JUMP, which moves to another state without going on to the next word of input; PUSH X, which calls the network recursively beginning at state X; and POP, which indicates the end of processing the current level and specifies a schema for building a piece of tree structure from the contents of the registers.

The actions on the arcs are: (SETR X Y), which replaces the contents of register X by the value of Y; (ADDR X Y), which adds the value of Y to the contents of register X without destroying the old value; (GETF X) which returns the value of the syntactic feature X associated with the current word; and (ABORTIF (NOT (DETAGREE))) which blocks the arc if the determiner does not agree with the head noun of a noun phrase (as in "a rocks"). Other actions not shown in the example can access previous register contents and test arbitrary predicates in order to perform some actions conditionally. The abort option is particularly useful for detecting errors in the input and blocking the analysis.

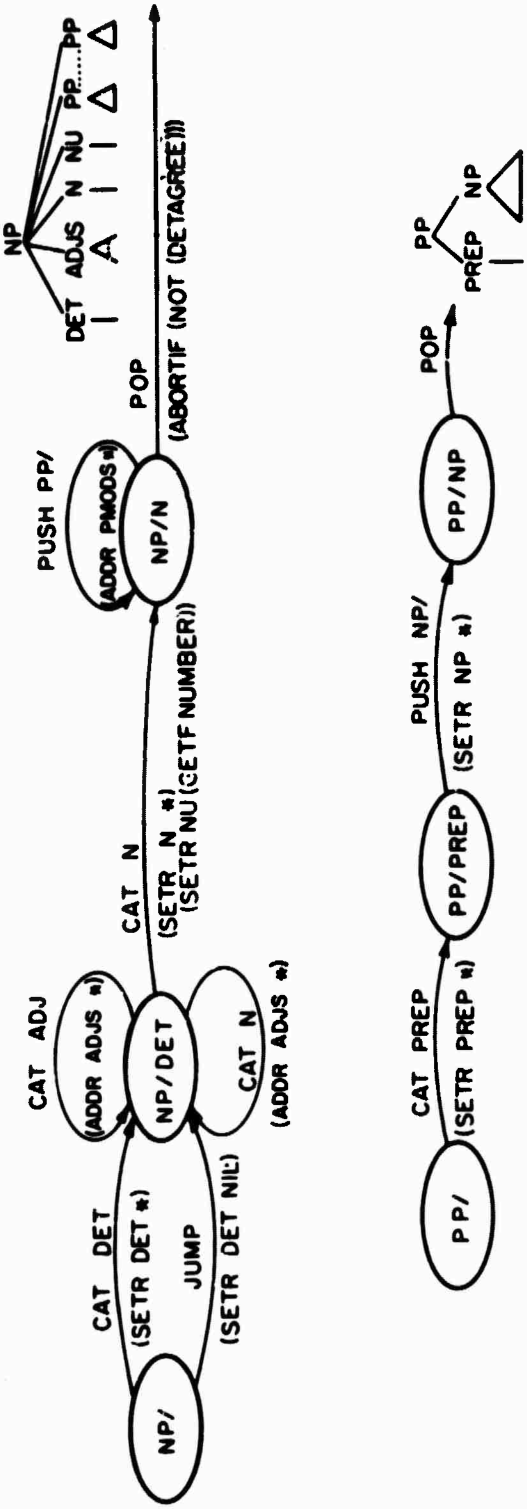


Figure 1. Small transition network grammar

The symbol * is used to refer to the current word of input, or, on a PUSH arc, to the tree structure returned by the recursive call. When operated as a text parser, the TNG mechanism is top down.

Several changes have been made in the form of Woods's original grammar formalism to adapt it to the speech environment. They are:

(1) The test portion of each arc, which used to be any LISP form, has been split into two tests (each of which is a LISP form). One test is context free, i.e. is concerned only with the current word of input, and can check syntactic features of that word. The other test is context sensitive and can check contents of registers which were set on previous arcs in the parse path. (For example, it can check number agreement between a determiner and head noun of a noun phrase to screen out such strings as "those trip.") This allows context free checks to be done as soon as possible, while the other test must wait until sufficient context has been established.

(2) The SENDR mechanism, which was originally developed to allow communication between constituent levels when parsing, has been eliminated. This is because it is convenient, almost necessary, for a speech parser to develop small constituents in isolation, without regard to the context of the constituent. If a word were passed down from a higher constituent to a lower one it would become an integral part of the parsing at that level. If another word were to be hypothesized in its place, the work of parsing the lower level would have to be redone. Thus it is useful to be able to parse, say, relative clauses such as "that I gave to you" without the presence of the context "the book that I gave to you".

Instead of using SENDR's, the grammar is arranged so that when a word is needed which formerly would have been sent down via a SENDR, a dummy node, e.g. **NP**, is used instead. Thus a constituent may be built which looks like:

```
S REL
  S NP PRO I
         FEATS NU SG
    AUX TNS PAST
    VP V GIVE
       NP **NP**
       PP PREP TO
          NP PRO YOU
             FEATS NU SG
```

The PUSH arc which looks for this constituent must then substitute whatever information would have been pushed down to fill in the place of the dummy node, and do whatever agreement checks are necessary. The constituent with its dummy node is placed in the well-formed-substring table so that it can be used, without reparsing, by any other process looking for a relative clause at that position.

(3) The HOLD list mechanism has been eliminated. The HOLD list was designed to handle the phenomenon known in transformational grammar as left extraposition -- the movement of a subpart of a constituent to a position above and to the left of the deep structure position (as in the fronting of question words: "What did he do that for?"). Putting an item on the HOLD list was like setting a global register which all lower levels could access. Since the HOLD list could be replaced by using SENDR's to send down information every time a PUSH was done, it can also be replaced by the use of a dummy symbol as described above.

(4) The LIFTR mechanism has been replaced. The LIFTR mechanism was analogous to SENDR except that it sent register information up to a higher level. This provides a way to pass information up which does not have a place in the syntactic structure at the current level. For example, one might want to pop a number as the structure (NUMBER 11510) with the feature DIGITS to indicate that it had been parsed from "one one five one oh" instead of "eleven thousand five hundred and ten". This would be useful since, if the number were to be interpreted as an account number rather than as a number of dollars, it would almost invariably be said in the former way.

> This capability has been retained, but in a different form. A special register may be set at any time during the parsing of a constituent to contain information which should be passed up. When the constituent is complete, the content of this register is attached to the constituent in the well-formed-substring table as its feature list. A PUSH arc may manipulate these features in any way, including using it in the structure at the higher level or putting some information in the special register at the higher level in order to pass it up again.

C. The Scope of the Grammar

The scope of the speech grammar has been extended from the very small grammar (11 states) with which we began in 1971 to a grammar of 70 states with almost the full power of the LUNAR grammar. Of course, some capabilities of the LUNAR grammar are not needed in the speech grammar, such as the ability to deal with punctuation. The grammar can currently handle declarative, imperative, and question sentences, with sentential complements and relative and reduced relative clauses. We have also included subgrammars to parse numbers and money expressions (e.g. "He spent 50 K," "The actual cost of the trip was three hundred fifty four dollars and nineteen cents," "The account is 11510").

The section of this report dealing with the grammatical characteristics of the travel budget domain [see IV.B.] gives a more detailed discussion of the capabilities of the current grammar.

D. Problems in Parsing Speech

Parsing speech is a much more difficult problem than parsing text. Because speech is continuous, word and sentence boundaries are usually obscured. Also, inaccurate or hasty articulation and the normal variation in the pronunciation of phonemes cause the pronunciation of a word in context to be very different from that in isolation. Acoustic processing results in uncertainty in the identification of phonemes and, therefore, of words -- especially small function words such as "the", "a", "of", "have", "did", etc. (Even if the acoustic component could identify phonemes uniquely, some ambiguity would be inevitable because of the occurrence of homonyms, and because word boundaries may be shifted, as in "tea meeting/team eating/team meeting".) In text processing there is no such inherent ambiguity, but any speech understanding system must be able to deal with it.

The implication for parsing is that the input to a parser for speech cannot be a string of uniquely determined words but must be something like a lattice of words (see Figure 2 for a word lattice for the first few milliseconds of the utterance "List all the samples which contain silicon"). When the parser wants the "next word" of the input it must be able to deal with a list of possible words and must be prepared to cope with the possibility that the right word is not included in that list.

It may also be the case that no usable word can be found at one or more places in the utterance, so the parser must also be able to deal with gaps in its input.

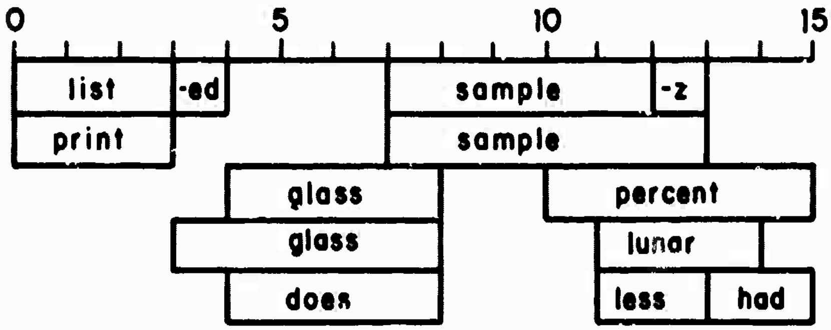


Figure 2. A partial word lattice

When processing text, a parser could reasonably take advantage of a number of extra-linguistic indicators such as punctuation marks (a period to delimit a sentence, commas to disambiguate certain complex conjunction constructions, etc.), capitalization (to indicate the start of a sentence or to distinguish proper nouns such as "Pat" from other words such as the verb "pat"), italics, underlining, quotation marks, and parentheses. (To illustrate the importance of these factors to comprehension, consider the following grammatical but unpunctuated string: "that which is is that which is not is not is not that so"). All of these cues are missing in speech. They are compensated for by the use of pauses, stress, changes in duration, pitch, and loudness, and other prosodic features. Unfortunately the current lack of knowledge about the acoustic correlates of prosodic features makes it almost impossible to use this rich source of information in speech understanding systems, so current speech parsers must cope with the increased ambiguity resulting from this lack of information.

1. The Purpose of Syntax

In most systems which work with natural language the purpose of the parser is to provide a representation of the syntactic units of the input and their relationships to one another. This representation is frequently a "deep structure" tree (as in Figure 3) which may then undergo semantic analysis or interpretation. The creation of a self-contained syntactic structure is not absolutely mandatory if enough semantic and interpretive processing is done together with the parsing, but in any case the syntactic component must be able to confirm that the input is grammatically correct, and we will assume that some structure for it is also produced. A parser for speech, however, must do more than this. In addition to detecting syntactic ambiguities (e.g. "I gave her cat food."), syntax must aid in selecting a syntactically well-formed sequence of words from the many sequences of words which are possible in the word lattice.

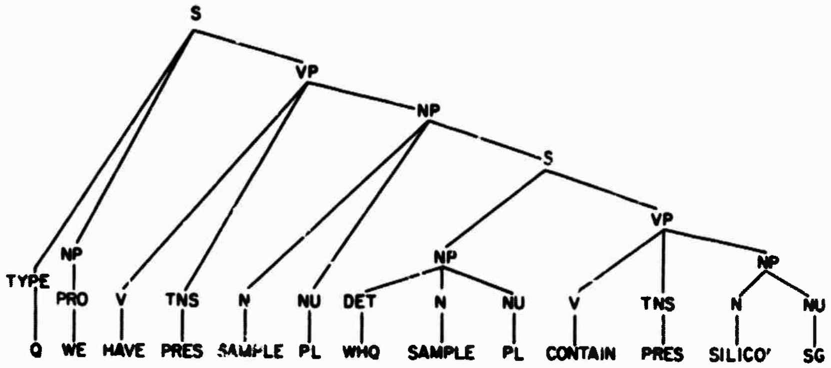


Figure 3. A deep structure for
"Do we have samples which contain silicon?"

Text parsers are designed on the assumption that the words given as input will form a grammatical sentence, so the duty of the parser is merely to determine the structure(s) of the sentence. A speech parser, however, must know that some (in fact, many) of its potential input sequences will be ungrammatical, and it must be able to detect and reject those sequences as early as possible.

Another goal of any speech parser must be to predict words or syntactic categories which could fill gaps in the word lattice. The type and correctness of the predictions which can be made depend on the nature of the grammar being used and the amount of context which is taken into account when making the predictions.

2. Existing Models

Assuming that the extensive body of work which has been done in the analysis of text has something to offer for the analysis of speech, let us examine two of the techniques which have been used. For a more complete description of these methods see the book by Aho and Ullman [1].

Top down methods of parsing (so called because they construct the deep structure tree by beginning at the root node and working down) are left-to-right and usually predictive; they begin by searching for a component of a given type and operate recursively, trying all possible ways of building the constituent before failing. The ability of this method to predict, at any point, the set of acceptable constructions which could appear in the input as a function of the context to the left is its strongest advantage. In speech analysis, the predictions may be used to eliminate some of the possible "next words" in the word lattice. This method has the disadvantage that if there is an error at or near the beginning of the input, the parser may not only take a long time to fail but will consider the last portion of the string only in the context of the earlier (erroneous) part. Thus little if any useful information may be gained about the structure of the last part of the input. Unless great care is taken to prevent duplication of effort when re-parsing portions of the input (by the use of a well-formed-substring table or by compacting methods such as

Earley's algorithm [1,14]), the lexical ambiguity of speech input could cause an exponential increase in the amount of work required.

Bottom up techniques such as Cocke's algorithm [1] begin with the leaves of an analysis tree and work up. First, all possible substrings of length one are considered and all one-word constituents formed. Then using this information all pairs of adjacent words are considered and all two-word constituents are formed. Then all adjacent three-, four-, five-,... word substrings are considered until the length of the string is reached. This method is neither left-to-right nor right-to-left and has the advantage of working with isolated sections of th input so that an error at one point will not prevent a correct analysis of another portion of the string. It unfortunately requires that all possible parsings of all sections of the input be found in parallel -- a procedure which is enormously wasteful of space and time even when a single string is being processed. The multiple words produced by an acoustic analyzer and lexical retriever together with the multiple syntactic categories for many of those words and the multiple ways they can be syntactically combined when only very local context is used exacerbate the problem to such an extent that a totally bottom up speech parser would be unthinkably slow.

What is needed is a scheme which can merge top down techniques with bottom up ones to combine directed, predictive analysis with immunity to errors in non-local context. The formalism of a transition network grammar (described in Section VI.B above) seems particularly well suited to such adaptation, for the following reasons. TNG´s allow easy prediction to both the right and left of any word of input. They are constructed in such a way that ambiguous information is separated only in the truly ambiguous part, allowing merging of the rest of the analysis. Some relief from contextual errors can be gained by limiting the context of any word in the input to only those words which may be in the same constituent. Finally, although TNG´s were designed to drive a parser in top down mode, bottom up information is easily accessible.

E. The BBN Speech Parser

Though the parser for the BBN speech understanding system uses an augmented transition network grammar (with the modifications described in Section VI.B), it is completely different in organization and operation from that of the LUNAR system.

The main features of the parser are:

(1) It is designed to start parsing anywhere in the input stream and to parse despite the lack of certainty as to the exact nature of the words at each point in the input.

(2) Complete constituents, when found, are stored in a well-formed-substring table (WFST) along with their features, boundaries, and a semantic evaluation of their meaningfulness so that they may be used by any other parse path which needs a constituent of that type at the same place without reparsing.

(3) As partial parse paths are built up, their pieces are also stored in tables so that any other parse which can use them need not reparse common sections of input.

(4) Using the grammar, the parser can make predictions about the words or syntax classes which could be used to extend a sequence of words in a theory either to the right or to the left. If a gap between words is small enough to contain just one word, the parser can predict just the class or classes of words to fill the gap.

(5) The control structure of the parser can be modified fairly easily to experiment with various combinations of backup, sequential, and parallel search. Currently, it uses a combination of depth first and breadth first techniques, usually following a single path but splitting into parallel paths when desirable.

(6) Care has been taken to allow the parser to interact frequently and easily with other components of the system (notably Semantics) in order to receive guidance and to verify completed constituents. Several aspects of the Syntax-Semantics interaction are discussed in Section VII.

(7) Although at any given moment the parser is concerned with only one theory, its data base contains all the information it has discovered in processing previous theories, thus allowing considerable sharing of information without duplication of effort. This organization allows for the occurrence of some event (such as the completion of a constituent) to alert the control component to the fact that certain previously processed theories may be affected by the event and should be queued for further processing.

1. Description

The syntactic component of BBN's speech system is one of a number of processes which work together to understand an utterance. For an overview of the entire system, see [47] or Section I. of this report. Very briefly, the structure of the system may be described as follows. There are a number of components (Acoustics, Lexical Retrieval, Syntax, Semantics, Pragmatics, and Control) which are called into action under the direction of the control component. Acoustic, phonological, and lexical processes produce from the acoustic signal a lattice of word matches for words with a high lexical score, similar to that in Figure 2. Only words of two or more phonemes are placed in the lattice initially since smaller words tend to match well everywhere and flood the lattice.

The semantic component selects subsets of this lattice based on semantic relationships among the words. Such a subset (in the form of a word match list) is associated with semantic, pragmatic and (initially empty) syntactic information and is termed a THEORY. It is an hypothesis about the content of the utterance. For the remainder of this section, the term "theory" will be used to refer to the word match list alone as well as to the larger structure of which it is a part.

When a theory has been constructed to which Semantics can add no more words, it may be sent to Syntax for processing. The initial input to the parser, then, is a list of word matches. This list will probably not span the utterance; there will be islands of word matches with gaps between them. Each word match may represent either a single word with definite boundaries, a single word with "fuzzy" boundaries, a word together with possible inflectional endings, a group of words which have the same semantic associations, or a combination of any of the above. Using brackets to delimit word matches and numbers to indicate the boundaries in the word lattice, a typical theory for the utterance "List all the samples which contain silicon" might look like:

```
[list ]     [   sample    ]    [contain] [silicon]
[print]     [sample(-z)   ]
0      3    7         12     13  16       22       29
```

When the parser is given a theory to process, it processes the islands of word matches in the theory from left to right and attempts to create for each island the PATHs (sequences of TRANSITIONs and CONFIGURATIONs, defined below) which represent the ways in which the island of words might be accepted by the grammar if surrounded by some suitable context. Then Syntax tries to extend the theory by finding (in the word lattice) or predicting words or syntactic classes which would provide a context consistent with its analyses. When Syntax has finished processing a theory, it adds to the syntactic part of the theory the configurations and transitions used in its analysis and returns to Control a score which is a measure of the amount of syntactic information gained by the analysis.

Each configuration represents a state of the grammar which the parser could be in at a particular boundary point in the current theory. Each transition represents a change from one configuration to another by following an arc of the grammar. A transition contains information about the arc which it represents, the word or words used by the transition and the possible register contents resulting from execution of the actions on the specified arc. Since a given transition may have any number of transitions to its left (because different contexts may precede it), and since the actions on an arc frequently make use of the context to the left by looking at register sets, there may be a number of sets of possible register contents associated with the transition.

Syntax can create data objects called MONITORs, EVENTs, and PROPOSALs which represent instructions to Control. A monitor is a demon which is placed on a particular point in the word lattice. The monitor's job is to watch for a word possessing some specific characteristic (such as a particular part of speech) to be placed in the lattice at that point. If and when a monitor is activated, it creates an event, which is a record of the word which caused the event, the theory which caused the monitor to be set, and an instruction indicating which component to call to process the event. When an event is processed, a new theory is created from the old one by including the new word. Syntax can create events directly whenever it notices a word already in the word lattice which could be used to extend the theory it is processing. Monitors are passive in the sense that they merely wait for a word which can activate them to appear. They do nothing to cause such a word to be found. A proposal, on the other hand, is, as far as Syntax is concerned, a command which causes Control to activate the word match component to look specifically for a particular word or syntactic category (whose members are enumerated) at a particular place in the word lattice. If a word is found, the corresponding monitor will be activated and an event created.

In order to make this flow of data and the relationships among the various sources of data more clear, Figure 4 shows schematically the flow of the data types just discussed.

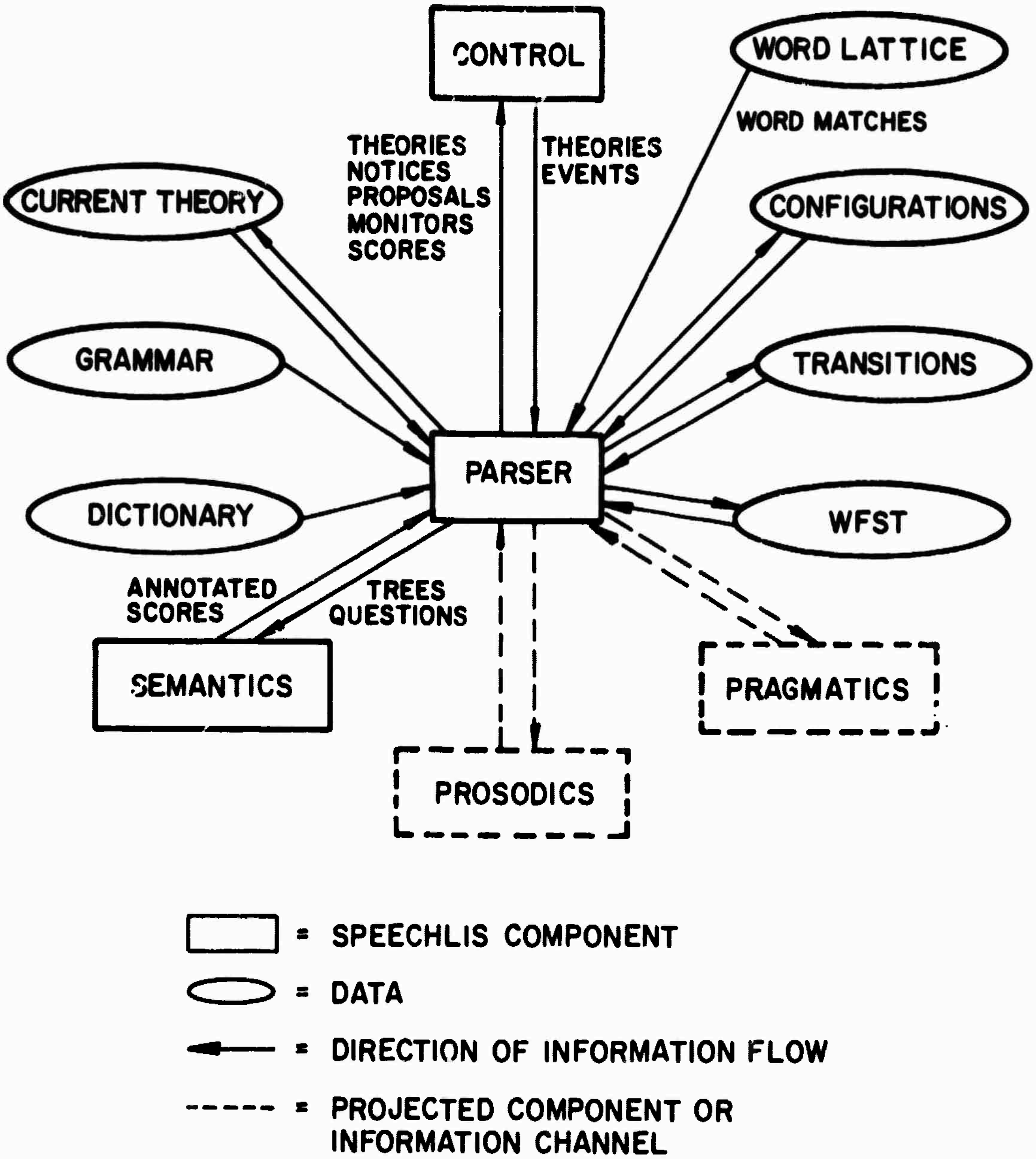


Figure 4. Data flow diagram for the speech parser

2. An Example

Working through a small example should help to explain the features of the parser and the data structures it builds. Consider the theory which was shown above. Figure 5 shows a map of some of the configurations (boxes) and transitions (arrows) which exist after the second island of the theory ("sample(s)") has been analyzed. The transitions are numbered in order of their creation and show the arc they represent and the sets of associated register contents. (The registers are not actually set until a path has been constructed from an initial configuration to a POP transition.) Let us assume that the semantic component had attached to the theory the constraint that "sample(s)" be used as a noun, not as a verb or as an adjective ("(he) samples the rocks", "(the) sample number"), Using this semantic restriction together with an appropriate index for the arcs of the grammar (refer to Figure 1), the parser can determine that the first CAT N arc from state NP/DET must be used to process the word "sample(s)" since the other CAT N arc actually uses the word as an adjective. In general there may not be semantic constraints on how the first word of an island can be syntactically realized, so all arcs would be found which could process the word as any of its possible parts of speech. Thus the parsing is begun in a bottom up mode.

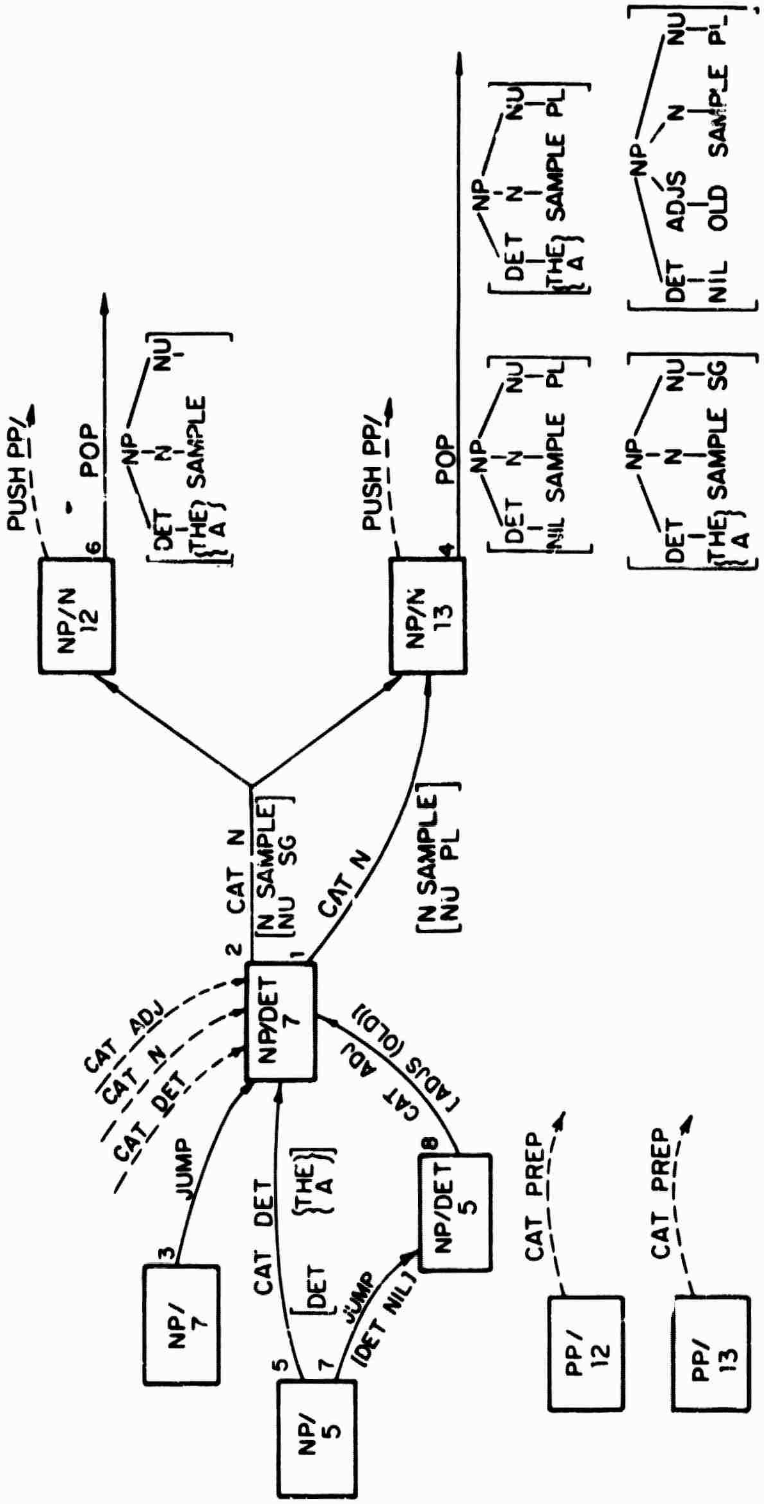


Figure 5. Map of transitions and configurations

Considering the plural possibility first, a transition is made from a configuration for state NP/DET at position 7 to a configuration for state NP/N at position 13. The singular case is "fuzzy" since the end position can be either 12 or 13, but the register contents will be the same in either case. Instead of creating two transitions with duplicate information, one transition (number 2) is created with multiple terminations. Multiple initial configurations are also permitted.

Now consider what could occur to the left of the island. Reference to the grammar shows that in order to get to state NP/DET the parser must take either the JUMP arc from NP/ or one of the CAT ADJ, CAT N, or CAT DET arcs. A transition for the JUMP arc can be created immediately since it needs no context. The word lattice is checked for the existence of a word of category ADJ, N, or DET and if one is found, an event relating it to the current theory is created. Whether or not such a word is found, monitors are set to watch the word lattice for an occurrence of a noun, adjective, or determiner at some later time. Syntax remembers the arcs which caused the monitors to be set and the configuration at that point (indicated by the dotted arrows in Figure 5) in order to be able to process an event should one occur.

Going back to our example, we have left open two configurations (NP/N at 12 and NP/N at 13) which may be considered for extension. All open configurations may be processed, but this results in many partial paths through the island. Actually they should be ordered according to the goodness of the paths which terminate on them. We are currently working on a formula for calculating a score for a path, based on such things as the length of the path, and perhaps even the lexical score of the words used. By trying to continue only the best-looking paths (but remembering the others), we cut down the number of possibilities which the parser must explore.

When a configuration is to be extended, the arcs from its state are tried one at a time in top down fashion. PUSH arcs, when encountered, cause an internal syntactic monitor to be set at a position in the parser's well-formed-substring table (WFST) where all constituents are placed when they are created. The PUSH arc also causes creation of a configuration for the state PUSHed to in order to begin processing for the constituent. If the end of the island has been reached, arcs which require context to the right of the island cause creation of events, monitors, and proposals just as they did on the left. In our example, this point is reached after the creation of configurations for state NP/N at positions 12 and 13 and the setting of monitors for prepositional phrases and prepositions. Whenever a path becomes blocked, a simple backup procedure is invoked to go back one step of the path and try another of the alternatives stored there.

Although this part of the parser is basically top down, it can be restricted by bottom up information. For example, whenever a word in an island is processed which Semantics has hypothesized must be used in a certain syntactic way, only the arcs of the grammar consistent with that hypothesis may extend the path through that word.

The rest of Figure 5 shows the transitions and new constituents which would be created for two events, one for the two determiners "the" and "a" and then one for the adjective "old". The test on the POP arc checks agreement between determiner and head noun and prevents noun phrases for "sample", "old sample", and "a samples" from being created.

A feature currently being designed for the parser will allow an action on any arc to be a call to Semantics to test the contents of various registers in order to determine whether or not that particular path appears to be semantically likely. For example, if the sequence "green zebra" is being processed with "green" as an adjective and the parser is considering the arc which would take "zebra" as the head noun, Semantics could be asked to determine how well the adjective fits the noun. Since the answer would be "not well at all", the parser could take this as an indication to lower the score for that path and try another possibility, such as the arc which would accept "zebra" as an adjective and look for another noun (e.g. "cage") to follow it.

Semantic guidance could be used to answer such questions as: "Given that a particular prepositional phrase has been found in the WFST and can be used to modify a particular noun, would the result be semantically meaningful?" or "A verb is about to be parsed, and the subject of the sentence is known. Could the noun phrase in the subject register actually serve as a subject of the verb?" Even pragmatic guidance could be used in a similar way ("Is it pragmatically likely that this verb is passivized?"), if it were known how to structure more pragmatic knowledge in a usable way.

Figure 5 shows part of the data base constructed for one theory only. As other theories are processed, they add to the same data base and may use the information already there. Thus, syntactic information may be shared across theories. This is especially important for the WFST, since once a constituent is placed there it is available to all other theories without re-parsing. Even partial paths may be shared, since once a configuration or transition has been created it is never duplicated but merely included in the syntactic part of any theory which can use it.

F. Conclusion

We have tried to show that one of the major problems facing a parser for speech is the lexical ambiguity of its input. The combinatorial possibilities induced by this ambiguity make straightforward applications of previous parsing techniques too lengthy and complex to consider.

We have attempted to reduce the combinatorial problem by the following methods: semantic and pragmatic pre-selection of small subsets of the total word lattice; the use of semantic guidance during parsing; a basically top down parsing algorithm with backup capabilities so that not all paths need be followed in parallel; a mechanism to allow ordering of the paths so that only the best are processed; merging of information whenever possible; use of the WFST to avoid re-parsing constituents which have already been found; and sharing syntactic information among theories to avoid re-parsing.

That these methods do substantially reduce the work required can be shown by an example which has been parsed by the system. The utterance was "How many samples contain silicon?" and the word lattice contained all the correct words as well as "give" in the same place as "how" and "any" in the same place as "many". Using a grammar of 43 states and 102 arcs, beginning with a theory for "sample(s) contain silicon", and processing an event for each of the other four words, it is estimated that a parser without the ability to share transitions and configurations among several theories, without backup, and without the WFST would create about 300 configurations and nearly 500 transitions. The BBN speech parser actually constructed a total of 104 configurations and 142 transitions. The parser was operating without semantic guidance or merged register information -- with these features a reduction in the number of transitions and configurations of about one third could be expected for this example.

Although we have come a long way toward building a parsing system for speech, there are still many things that need to be done. Probably the most important is to develop ways to take more syntactic context into account when scoring the parse paths and to start the scoring procedure during the construction of partial paths rather than waiting for complete constituents to be built. This would cut down even further on the combinatorial explosion of syntactic possibilities. More accurate scoring would also allow incorrect or very unlikely paths to be aborted earlier.

The grammar also requires work both to extend its capabilities and to tighten its constraints so that invalid sequences are detected and rejected as soon as possible. When parsing text one has the luxury of being able to assume that the input is a grammatical string, but in the speech environment one must assume that even if the sentence which was said is grammatical, there will be enough error in the acoustic and lexical processing to produce high scoring but incorrect (and frequently ungrammatical) sequences of words. By tuning the grammar to recognize errors, the parser will be more efficient in rejecting erroneous theories.

Although there is always more work which remains to be done, we have established a framework which will provide fertile ground for experimenting with various hypotheses concerning parsing strategies and syntactic processing. We expect the syntactic component to continue to serve as a tool to help us learn about the role of syntactic information in the environment of a total speech understanding system.

VII. ASPECTS OF SEMANTIC KNOWLEDGE FOR AUTOMATIC SPEECH UNDERSTANDING

A. Introduction

If a speech understander must use semantic knowledge to constrain the many possible ways of hearing an utterance, then his semantic knowledge must represent what can be meaningful and what may be expected at any point in a dialogue. Preferring a meaningful and likely utterance to one that is not, a speech understander must be able to use his semantic knowledge to seek one out. Thus the knowledge of what can be meaningful and the ability to make predictions based on that knowledge may be the most important aspects of semantics for speech understanding. As to the former, it is more important to know that physical objects can have color than that canaries are yellow. As to the latter, if the objects in a group can be distinguished by color, then it is reasonable to expect a color specification in identifying a subset of them. This makes "yellow birds", for example, a meaningful and likely phrase. This is not to say that factual knowledge is not useful in speech understanding, but rather, as we hope to show below, that it is just not as powerful an aid as other types of semantic knowledge. Let us now consider what types of semantic knowledge determine what is meaningful and enable predictions.

1. Knowledge of Names and Name Formation

Semantic knowledge of the names of familiar things and of models for forming new ones permits a listener to expect and hear meaningful phrases. For example, knowing the words "iron" and "oxide" and what they denote, and that a particular oxide (or set of them) may be specified by modifying the word "oxide" with the name of a metal, may enable a listener to hear the sequence "iron oxides", rather than "iron ox hides" or even "Ira knocks sides".

2. Knowledge of Lexical Semantics

Knowledge of lexical semantics (models of how words can be used and the correspondence between concepts in memory and their surface realizations) enables the listener to predict and verify the possible surface contexts of particular words. Along with the previously mentioned knowledge of names and name formation, this contributes to "local" recognition of an utterance: given a hypothesis that a word has occurred in the utterance, what words could have appeared to its left or right. For example, the concept of CONTAINMENT, invoked, *inter alia*, when the word "contain" appears in a sentence, has two other concepts strongly associated with it -- a container and a containee. (These might also be called the "arguments" to CONTAINMENT. Note that, in this report, concepts will be distinguished from words by being written in capital letters.) When "contain" is used in an active sentence, it must have a subject which is understood to be a location or container, and an object which is capable of being located or contained. In a passive sentence, the roles are interchanged: the active object becomes the passive subject and the active subject or location is realized in a prepositional phrase headed by "in". E.g.:

Every egg contains a yolk.
(Active)
A yolk is contained in every egg.
(Passive)

There are several things to notice here. First, given the possibility of being able to hear the initial segment of the first utterance as either "every egg" or "every ache", one would usually hear the former, since it is a more likely container, especially for yolks. Secondly, given that little words lose most of their phonetic identity in continuous speech and that in hearing the second utterance we have a strong hypothesis that it is of a passive sentence, we can use the knowledge of how "contain" passivizes to predict and verify the occurrence of "is" and "in" in the acoustic signal. If we cannot satisfy ourselves as to their existence in the utterance, we may decide to change our earlier hypothesis that the utterance was of a passive sentence.

Thirdly, while we can profitably use lexical semantics to predict the local context of a word by going to the concepts it can partially instantiate and predicting what can fill the gaps, it does not gain one much to make predictions about the way in which a completely uninstantiated concept will be realized. There are usually too many possibilities available. For example, the concept of CONTAINMENT comes across in all the following phrases:

Rocks containing sodium
Sodium-containing samples
Sodium-rich basalts
Igneous samples with sodium
Samples in which there is sodium
Rocks which have sodium

3. Knowledge of Conceptual Semantics

Knowledge of conceptual semantics, how concepts are associated in memory, contributes to a listener's ability to make "global" predictions across utterances, as well as ones local to a given one. The global predictions are primarily of the nature: if one concept is under discussion, which other ones are soon likely to come up and which ones not. Expectations about which related concepts need not be mentioned in the discourse help the listener accept and accommodate such discourse tricks as ellipsis and anaphora. A short example of conversation should suffice here to illustrate the point.

"I´m flying to New York tomorrow. Do you know the fare?"
"About 26 dollars each way."
"Do I have to make reservations?"
"No."
"Super."

There are several points to make here. First, the concept of a trip is strongly linked with such other concepts as destinations, fares, transportation mode, departure date, etc. So one might expect them to be mentioned in the course of a conversation about a trip. Secondly, the strength of these associations is both domain-, context- and user-dependent. If the domain concerns planning trips, as in making airline reservations, then destination and departure date would seem to have the strongest links with trips. In another domain such as managing the travel budget for a company, it may only be the cost of the trip and who is paying for it that have this strong association. As far as context and user dependency are concerned, the company accountant´s primary interest in business trips may be quite different from that of a project leader wondering which of his people are going where.

Thirdly, the places where ellipsis is most likely to occur seem to correlate well with strong inter-concept associations. This is useful information since it suggests when not to look hard for related concepts in the local context. For example, "the fare" and "reservations" are both elliptical phrases: "the fare" must be for some trip via some vehicle at some time. But fares are so strongly linked with these notions that is is not necessary to mention them explicitly as in, "Do you know the current air fare to New York?" Again, what the reservations are for is not stated explicitly, but must also be for the aforementioned flight. Without a knowledge of the concepts associated with trips and fares and how "strong" the links are, none of the above local or global predictions could be made. What's more, the above conversation would be incoherent. (N.B. Conceptual associations such as those discussed above are of course not the only source of "global expectations". Rhetorical devices available to a speaker who chooses to use them, such as parallelism and contrast, add to global expectations about the structure of future utterances. In addition, problem solving situations also have a strong influence on the nature of discourse and the speaker's overall linguistic behavior.)

4. Knowledge of the Use of Syntactic Structures

Knowledge of the meaningful relations and concepts that different syntactic structures can convey enables the listener to rescue cues to syntactic structure which might otherwise be lost. Among the meaningful relations between two concepts, A and B, that can be communicated syntactically are that B is the location of A, the possessor of A, the agent of A, etc. Also among syntactically communicated concepts are set restriction (via relative clauses), eventhood (via gerund constructions), facthood (via 'that'-complements), etc. Syntactic structure is often indicated by small function words (e.g. prepositions and determiners) which have very imprecise acoustic realizations. The knowledge of what semantic relations can meaningfully hold between two concepts in an utterance and how these relations can be realized syntactically can often help in recovering weak syntactic cues.

On the other hand, one's failure to recover some hypothesized cue, once attempted, may throw doubt on one's semantic hypothesis about the utterance. For example, the preposition "of" can practically disappear in an utterance of "analyses of ferrobasalts". Yet the only meaningful relation between "analyses" and "ferrobasalts" that can be expressed with this word order requires that "ferrobasalts" be realized as a prepositional phrase headed by "of" or "for". If one hypothesizes that something is an utterance of "analyses of ferrobasalts", and one is reasonably certain only that he has heard "analyses" and "ferrobasalts", he can try to confirm the occurrence of one of these prepositions in the speech signal. If he can, it is more believabl that "analyses of ferrobasalts" was the intended sentence. If he cannot, it becomes doubtful, though not impossible, that "analyses" and "ferrobasalts" really did occur in the utterance. An alternative hypothesis, for example, that the intended sentence was "analyses for all basalts", may become more likely.

5. Knowledge of Specific Facts and Events

Knowledge of specific facts and events can also be brought in as an aid to speech understanding, though it is less reliable than the other types of semantic knowledge discussed above. This is because it is more likely for two people to share the same sense of what is meaningful than for them to be aware of the same facts and events. Fact and event knowledge can be of value in confirming, though not in rejecting, one's hypotheses about an utterance. For example, if one knows about Dick's recent trip to Rhode Island for the America's Cup, and one hears an utterance concerning some visit Dick had made to -- Newport?, New Paltz?, Norfolk?, Newark? -- one would probably hear, or choose to hear, the first, knowing that Dick had indeed been to Newport. However, one couldn't reject any of the others, on the grounds that the speaker may have more information than the listener.

B. Studying Semantics in the Context of Speech

We have argued above that speech understanding benefits from the use of semantics. We can also argue that semantics benefits from being studied in the context of speech. That is, in our speech research, we have become aware of aspects of the language understanding process that either have not arisen in the attempt to understand printed text, or have done so and been consciously put aside as not crucial to the level of understanding being attempted.

The first aspect concerns the nature of the input. In spoken language, as distinct from written text, word boundaries are not given unambiguously, and hence words may not be uniquely identified. Compounding the problem is the sloppy, often incomplete realization of each word. In addition, coarticulation phenomena are such that the correct identification of a word in the speech signal may depend on the correct identification of its neighbors. Conversely, a word's incorrect identification may confound that of its neighbors.

As a result of the nature of its input, understanding spoken language seems to require a special mode of operation, such as "hypothesize and test", in order to get around the vague, often incomplete, realization of each word in the utterance. That is, one needs the ability to make hypotheses about the content of some portion of the input and then verify that that hypothesis is consistent with a complete interpretation of the input. The same process must go on in the understanding of handwritten text, which is inevitably sloppy and ill-formed. Notice, for example, how the same scrawl is recognized as two different words in contexts engendering different predictions.

Pole vaulting was the third event of the meet.

After dinner, Jack went home.

Recently, researchers concerned with modelling human language understanding, notably Riesbeck [35], have also proposed this mode of operation, "parsing with expectations", as the way of getting directly to, in most cases, the "intended" interpretation of a sentence. His argument is that this model accounts for the fact that people do not even seem to notice sense ambiguities if they are expecting one particular sense.

A second point is one of degree. Although people have paid much attention to giving machines the ability to reject "bad" readings of a sentence while accepting "good" ones, the examples they have considered in this regard have been very gross and simple in comparison to some very subtle ones that arise in speech. For example, the problem of "bad" readings arising from incorrect modifier placement is one frequently discussed, e.g. rejecting the anomalous reading of

"I saw the Grand Canyon flying to New York."

in which the Grand Canyon is doing the flying. In understanding a speech utterance, whose acoustic realization is always vague and ambiguous, the problem of evaluating the "badness" or "goodness" of such possible readings as those shown below is much more subtle.

How many people like ice cream?
Do many people like ice cream?
Do any people like ice cream?
Do eighty people like ice cream?
Do many people, like I, scream?

Some are "better" than others: one is forced into weighing many factors in choosing the best -- closeness of some realization of the reading to the acoustic signal, appropriateness of the reading to the context, likelihood of the reading within the context, etc. And all the factors may not point to the same reading as being best.

The next point about the advantages of studying understanding in the speech context is that there are phenomena relevant to understanding which are found either exclusively in spoken language, or mainly there and only rarely in written text.

First there are the kinds of errors that frequently occur in speech wnich must be accounted for in any valid model of human language understanding. The errors occur at all linguistic levels -- phonemic, syntactic, ana semantic. Ones seemingly related to semantic organization (because the meaning of the resu. ing utterance seems close to the supposed intention of the speaker) include malapropisms, portmanteaus, mixed metaphors and idioms, etc. For example,

> "I´m glad you reminded me: it usually takes a week for something to sink to the top of my stack." ["sink in" ~ "rise to the top of the stack"]

> "Follow your hypothesis to its logical confusion." ["logical conclusion"]

(See [17] for additional examples.) These errors rarely occur in text, whose production is much more deliberate and considered than that of speech. Since they force a constraint on valid models of human semantic organization which correct linguistic behavior does not, they are valuable to study and can be, only in the context of speech.

Another of these phenomena is that of stress, intonation, and phrasing. Though many linguists would argue that they are regularly predictable on the basis of the syntactic structure of the utterance alone, I would agree with Bolinger [5] that these are not only syntactic phenomena, but are also used by a speaker to reflect his intended meaning and focus. Thus, to quote two of Bolinger´s examples, the difference in stress patterns between the two utterances shown below cannot be accounted for on the basis of syntactic structure, which is the same for both, but reflects a difference in information focus.

> The end of the chapter is reserved for problems to solve.

> The end of the chapter is reserved for problems to computerize.

"Computerize" is richer in meaning than simply "solve". The choice of the former verb, rather than the latter, seems to reflect a decision that the action, not the object (i.e. "problems") is the point of information focus. The difference in intonation reflects this choice.

There are two points here: first, it is possible in speech to have several different, but simultaneous, cues to the same information. For example, potential ambiguities in the scopes of prepositional phrases may never arise because of semantic constraints or contextual knowledge or appropriate intonation or phrasing. It is an interesting question whether or not a speaker actually uses all possible cues if fewer will suffice to resolve a potential ambiguity. More generally, there are factors which any model of human language understanding must account for, like the ones above, which can only be studied in the context of speech.

Finally, the attempt to understand speech forces us to confront and deal with what we consider one of the most important and difficult to understand aspects of any decision process, and that is the role of error analysis and correction. We mentioned earlier the inherently ambiguous nature of the input. Given that we have decided that our reading of part or all of an utterance must be wrong, we must be able to suggest where the source of the error lies and what the best alternative hypothesis is. Moreover we must do so efficiently, lest we fail to come up with a satisfactory reading in reasonable time. These problems of error analysis and correction have been the focus of a great deal of past, present and future research in Artificial Intelligence, research which is being avidly followed by the speech understanding community. (See [30,43,44] for several different schemes for dealing with these problems.)

C. Specific Semantic Problems in Speech Understanding

We shall now discuss in more detail the position of semantics in SPEECHLIS, in terms of how a speech understander might use a knowledge of meaningful concepts and their possible surface realizations in order to recover a speaker's intended utterance. Before doing so though, we will present a brief description of SPEECHLIS from the point of view of its semantic component, so as to see the kinds of information available for making and verifying semantic hypotheses.

1. The SPEECHLIS Environment

An initial, usually large, lattice of good big word matches [see Chapters 1 and III] serves as input to the syntactic, semantic, and pragmatic components of the system. Subsequent processing involves these components working, step by step, both separately and together, to produce a meaningful and contextually apt reconstruction of the utterance, which is hoped to be equivalent to the original one. Steps in proposing or choosing a word reflect some hypothesis about what the original utterance might be. In SPEECHLIS, this notion of a current hypothesis is embedded in an object we call a theory, which is specifically a hypothesis that some set of word matches from the word lattice is a partial (or complete) reconstruction of the utterance. Each step in the higher-level processing of the input then is the creation, evaluation, or modification of such a theory.

The word lattice is not confined, however, to the initial set of "good, long" word matches. During the course of processing, any one of the higher level components may make a proposal, asking that a particular word or set of words be matched against some region of the input, usually adjacent to some word match hypothesized to have been in the utterance. The minimum acceptable match quality in this case would be less than in the undirected matching above for two reasons. First, there would be independent justification from the syntax, semantics, and/or pragmatics components for the word to be there, and second, the word may have been pronounced carelessly because that independent justification for its existence was so strong. For example, take a phrase like "light bulb", in ordinary household conversation. The word "light" is so strongly predicted by bulb in this environment, that its pronunciation may be reduced to a mere blip that something preceded "bulb". In the case of proposals made adjacent to, and because of, some specific word match, the additional information provided by the phonetic context of the other word match will usually result in a much different score than when the proposed word is matched there independent of context.

The control component governs the formation, evaluation, and refinement of theorics, essentially deciding who does what when, while keeping track of what has already been done and what is left to do. It can also take specific requests from one part of the system that another part be activated on some specific job, but retains the option of when to act on each request. (In running SPEECHLIS with early versions of the control, syntactic and semantic components, we found several places where, for efficiency, it was valuable for Syntax to be able to communicate directly with Semantics during parsing, without giving up control. (N.B. We will be using initial capitals on the words "syntax", "semantics" and "pragmatics" when referring to parts of SPEECHLIS.) Thus, it is currently also possible for Syntax to make a limited number of kinds of calls directly to Semantics. How much more the initial control structure will be violated for efficiency's sake in the future is not now clear.)

The reason that processing does not stop after initial hypotheses have been formed about the utterance is that various events may happen during the analysis of a theory which would tend to change SPEECHLIS's confidence in it, or cause SPEECHLIS to want to refine or modify it. For example, consider some utterance extracted from a discussion of the lunar rocks. Under the hypothesis that the word "lunar" occurred in the utterance, a good match found for "sample" to its right would only increase our confidence that both words were actually there in the original utterance. An entity called an Event Monitor can be set up as an active agent during the processing of a theory by some higher-level component, to watch for some particular event which would change that component's opinion of the theory. When such an event has occurred, the monitor would create an appropriate notice. Notices are sent to the control component which decides if and when to act on them. Only when a notice is acted upon will the appropriate revaluation, refinement, or modification occur. Examples of semantic monitors and events will be found later on in this chapter.

To summarize then, the semantics component of SPEECHLIS has available to it the following facilities from the rest of the system: access to the words which have been found to match some region of the acoustic input, and information as to how close to the description of the input that match is ability to ask for a word to be matched against some region of the input and ability to build or flesh out theories based on its own knowledge and to study those parts of a theory built by Syntax and Pragmatics. Given this interface with the rest of the SPEECHLIS world, how does Semantics make its contribution to speech understanding?

2. How SPEECHLIS Semantics Works

The primary source of permanent semantic knowledge in SPEECHLIS is a network of nodes representing words, "multi-word names", concepts, specific facts, and types of syntactic structures. A network representation was chosen because the local and global semantic predictions about an utterance described earlier come from the associations among words and concepts in the domain and their possible surface realizations. Associated with each concept node is a data structure containing further information about its relations with the words and other concepts it is linked to, and which is also used in making predictions. The following sections describe how such predictions are enabled.

a. Network-based Predictions

(1) Multi-Word Names

Each content word in the vocabulary (i.e. words other than articles, conjunctions, and prepositions; for example "ferric", "iron", "contain") is associated with a single node in the semantic network. From each word node, links go out to various other nodes. The first links of interest in considering local predictions are those that go to nodes representing "multi-word names" of which the original word is a part. For example, "fayalitic olivine" is a multi-word name linked to both "fayalitic" and "olivine"; "fine-grained igneous rock" is one linked to the word "fine-grained" and the multi-word name "igneous rock".

Representing multi-word names in this way enables us to maintain a reasonable size dictionary in SPEECHLIS (i.e. by not having to make up compound entries like "fayalitic-olivine" and "principal-investigator") and also to make local predictions. That is, any given word match may be partial evidence for a multi-word name of which it is a part. The remaining words may be in the word lattice, adjacent and in the right order, or missing due to poor match quality. In the former case, one would eventually notice the adjacency and hypothesize (i.e. create a theory) that the entire multi-word name occurred in the original utterance. In the latter case, one would propose the missing words in the appropriate region of the word lattice, with a minimum acceptable match quality directly proportional to the urgency of the success of the match. That, in turn, depends on how necessary it is for the word in the match to be part of a multi-word name. That is, given a word match for "oxide", Semantics would propose "ferrous" or "ferric" to its left, naming "ferrous oxide" or "ferric oxide". Given a match for "ferric" or "ferrous", Semantics would make a more urgent proposal for "oxide", since neither word could appear in an utterance alone. Further details on the proposing and hypothesizing processes will be given below.

There is another advantage to representing multi-word names in this way rather than as compound entries in the dictionary. As an immediate consequence, it turns out that fayalitic olivine is a type of olivine, a fine-grained igneous rock is a type of igneous rock which is a type of rock, and a principal investigator is a type of investigator. No additional links are needed to represent this class information for them.

(2) Concept-Argument Relations

From the point of view of Semantics, an action or an event is a complex entity, tying several concepts together into one that represents the action or event itself. Syntactically, an action or event can be described in a single clause or noun phrase, each concept realizing some syntactic role in the clause or phrase. One of these concepts is that associated with the verb or nominal (i.e. nominalized verb) which names the relation involved in the action or event. The other concepts serve as arguments to the relation. For a verb, this means they serve as its subject, object, etc.; for a nominal, it means they serve as pre-modifiers (e.g. adjectives, noun-noun modifiers, etc.) or as post-modifiers (e.g. prepositional phrases, adverbials, etc.). For example,

```
John went to Santa Barbara in May.
SUBJ VERB     PREP PHRASE   PREP PHRASE

John´s  trip to Santa Barbara in May.
PREMOD NOMINAL   PREP PHRASE  PREP PHRASE
```

In the semantic network, an action or event concept is linked to the one which names the relation and the ones which can fill its arguments.

Semantics uses its knowledge of words, multi-word names, and concepts to make hypotheses about possible local contexts for one or more word matches, detailing how the word matches fit into that context. Given a word match, Semantics follows those links in the network which lead from the word to concepts of which it is an instance, and also to multi-word names and concepts which it may partially instantiate. On each of the nodes which represent other components of the partially instantiated name or concept, Semantics sets an event monitor. In following network links for another word match, should a monitored node be instantiated (and conditions on the instantiation specified in the monitor be met), an event notice would be created, calling for the construction of a new, expanded theory.

To see this, consider the network shown in Figure 1 and a word match for "oxide". Since "oxide" occurs in the multi-word names "ferrous oxide" and "ferric oxide", Semantics would set monitors on the nodes for "ferrous" and "ferric", watching for either's instantiation to the immediate left of "oxide". It would also propose them there. Since the net shows that oxides can be constituents of rocks and a rock constituent can be one argument to the concept CONTAIN (the other argument being the concept SAMPLE), Semantics would also set a monitor on the node for CONTAIN and one on the node for SAMPLE.

If Semantics is later given a word match for "contain" or one of its inflected forms, or one which instantiates SAMPLE (e.g. "rock"), it would be seen by the appropriate monitor when it reached the node for CONTAIN (or SAMPLE), and result in the creation of an event notice linking "oxide" with the new word match.

SMALL SEMANTIC NETWORK

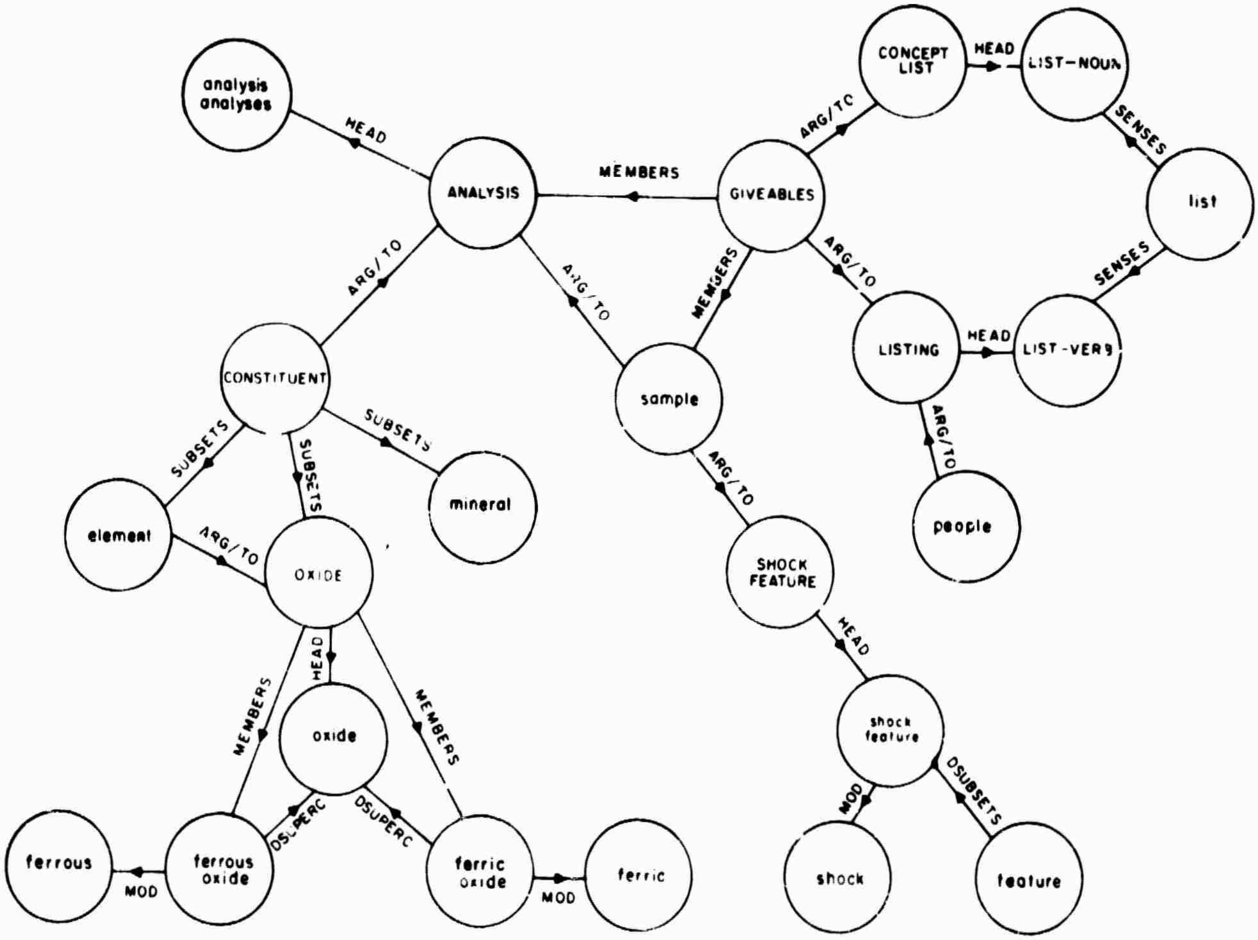


Figure 1.

Each notice has a weight representing how confident Semantics is that the resulting theory is a correct hypothesis about the original utterance. In the above, Semantics is less certain that a theory for "rock" and "oxide" will eventually instantiate the concept CONTAIN than it 's that a theory for "contain" and "oxide" will do so. (That is because there are many other possible ways of instantiating both SAMPLE and CONSTITUENT, but only "contain" or one of its inflections can instantiate the head of CONTAIN.) The event for the latter is therefore given a higher weight than the former.

(3) Syntactic Structures

Nodes corresponding to the syntactic structures produced by the grammar (e.g. noun phrases, to-complements, relative clauses, etc.) are also used in making local predictions. First, if an argument to some concept can be specified as a particular syntactic structure with a particular set of syntactic features, we want to predict an occurrence of that structure, given an instantiation of the concept's head. For example, a concept headed by "anticipate" may have as its object an embedded sentence whose tense is future to the tense of "anticipate".

> I anticipated that we would have made 5 trips
> to L.A. by November.

We want to be able to predict and monitor for any such structures and notice them if built.

More generally, we want to be able to use any co-occurrence restrictions on lexical items and syntactic structures or features in making predictions. For example, when different time and frequency adverbials may be used depends on the mood, tense, and aspect of the main clause and certain features of the main verb. "Already", for instance, prefers that clauses in which it occurs, headed by a non-stative verb, be either perfective or progressive or both, unless a habitual sense is being expressed. E.g.

John has already eaten 15 oysters.
John is already sitting down.
?John already ate 15 oysters.
(Perfective is preferable.)
*John already sits down.
John already runs 5 miles a day. (Habitual)

Secondly, if a concept with an animate agent as one of its arguments is partially instantiated, Semantics might want to predict some expression of the agent's purpose in the action. Now it is often possible to recognize "purpose" on syntactic grounds alone, as an infinitive clause introduced by "in order to", "in order for X to", "to" or "for X to". For example,

John's going to Stockholm to visit Fant's lab.
I need $1000 to visit Tbilisi next summer.
John will stay home in order for Rich to finish his paper.

These syntactic structure nodes then facilitate the search for a "purpose": they permit monitors to be set on the semantic concept of PURPOSE, which can look for, _inter alia_, the infinitive clauses popped by Syntax.

b. Case Frame based Predictions

(1) Description of a Case Frame

Additional information about how an action or event concept made up of a relation and its arguments may appear in an utterance is given in a <u>case frame</u>, a la Fillmore [16], associated with the concept. Case frames are useful both in making local predictions and in checking that some possible syntactic organization of the word matches in a theory supports Semantics' hypotheses. Figure 2 shows the case frames for the concepts ANALYSIS and CONTAIN.

A case frame is divided into two parts: the first part contains information relating to the case frame as a whole: the second, descriptive information about the cases. (In the literature, cases have been associated only with the arguments to a relation. We have extended the notion to include the relation itself as a case, specifically the head case (NP-HEAD or S-HEAD). This allows a place for the relation's instantiation in an utterance, as well as the instantiations of each of the arguments.)

Among the types of information in the first part of the case frame is a specification of whether a surface realization of the case frame will be parsed as a clause or as a noun phrase, indicated in our notation as (REALIZES . CLAUSE) or

(REALIZES . NOUN-PHRASE). If it is parsed as a clause, further information specifies which cases are possible active clause subjects (ACTIVSUBJ's) and which are possible passive clause subjects (PASSIVSUBJ's).

CASE FRAME FOR ANALYSIS

```
(((REALIZES . NOUN-PHRASE))
  (NP-HEAD (EQU .14) NIL OBL)
  (NP-OBJ (MEM .1) (OF FOR) ELLIP)
  (NP-LOC (MEM .7) (IN FOR OF ON) ELLIP))
```

(a)

CASE FRAME FOR CONTAIN

```
(((REALIZES . CLAUSE)
  (ACTIVSUBJ S-LOC)
  (PASSIVSUBJ S-PAT))
  (S-HEAD (EQU.20) NIL OBL)
  (S-LOC (MEM .7) (IN) OBL)
  (S-PAT (MEM .1) NIL OBL))
```

(b)

CONCEPT 14 - CONCEPT OF ANALYSIS
CONCEPT 1 - CONCEPT OF COMPONENT
CONCEPT 7 - CONCEPT OF SAMPLE
CONCEPT 20 - CONCEPT OF CONTAIN

Figure 2.

In the case of CONTAIN (Figure 2b), the only possible active subject is its location case (S-LOC), and the only possible passive subject is its patient case (S-PAT). For example,

```
    Does each breccia contain olivine?
              S-LOC          S-PAT

    Is olivine contained in each breccia?
        S-PAT                    S-LOC
```

(While not usual, there are verbs like "break" which allow several possible cases to become its active subject.

> John broke the vase with a rock.
> A rock broke the vase.
> The vase broke.

However, which case actually does so falls out from which cases are present. In ACTIVSUBJ, the cases are ordered, so that the first one which occurs in an active sentence will be the subject. There is no syntactic preference, however, in selecting which case becomes passive subject, so the case names on PASSIVSUBJ are not ordered.) The first part of the case frame may also contain such information as inter-case restrictions, as would apply between instantiations of the arguments to RATIO (i.e. that they be measurable in the same units).

The second part of the case frame contains descriptive information about each case in the frame.

(a) its name, e.g. NP-OBJ, S-HEAD (The first part of the names gives redundant information about the frame´s syntactic realization: "NP" for noun phrase and "S" for clause. The second part is an abbreviated Fillmore-type [16] case name: "OBJ" for object, "AGT" for agent, "LOC" for location, etc.)

(b) the way it can be filled - whether by a word or phrase naming the concept (EQU) or by either´s naming an instantiation of it (MEM), e.g. (EQU . SAMPLE) would permit "sample" or "lunar sample" to fill the case, but not "breccia". Breccia, by referring to a subset of the samples, only instantiates SAMPLE but does not name it.

(c) a list of prepositions which could signal the case when it is realized as a prepositional phrase (PP). If the case were only realizable as a premodifier in a noun phrase or the subject or unmarked object of a clause, this entry would be NIL.

(d) an indication of whether the case must be explicitly specified (OBL), whether it is optional and unnecessary (OPT), or whether, when absent, it must be derivable from context (ELLIP). For example, in "The bullet hit.", the object case - what was hit - must be derivable from context in order for the sentence to be "felicitous" or well-posed. (We plan to replace this static, three-valued indication of sentence level binding with functions to compute the binding value. These functions will try to take into account such discourse level considerations as who is talking, how he talks and what aspects of the concept he is interested in.)

(2) Uses of Case Frames

Semantics uses case frame information for making local predictions and checking the consistency of syntactic and semantic hypotheses. These predictions mainly concern the occurrence of a preposition at some point in the utterance or a case realization's position in an utterance relative to cases already realized. The strength of such a prediction depends on its cost: the fewer the words or phrases which could realize the case, and the narrower the region of the utterance in which to look for one, the cheaper the cost of seeking a realization. Since there are fewer words and phrases which name a concept (EQU marker) as opposed to instantiating it (MEM marker), cases marked EQU would engender stronger predictions. The <u>urgency</u> of the prediction depends on its likelihood of success, given that the hypothesis is true: if the case must be realized in the utterance (OBL marker), the prediction should be successful if the initial hypothesis about the concept associated with the case frame is correct. If the case need not be present in the utterance (ELLIP or OPT marker), even if the initial hypothesis is correct, the prediction need not be successful.

Consider the case frame for ANALYSIS in Figure 2a for example. If we were to have a theory that the word "analysis" occurred in the utterance, we would predict the following: 1) an instantiation of either COMPONENT or SAMPLE to its immediate left (that is, as a premodifier), 2) either "of" or "for" to its immediate right, followed by an instantiation of COMPONENT, and 3) either "in", "for", "of", or "on" to its immediate right, followed by an instantiation of SAMPLE. It doesn't matter that the above predictions are contradictory: if more than one prediction were successful (i.e. there were more than one way of reading that area of the speecn signal), it would simply be the case that more than one new theory would be created as refinements of the original one for "analysis", each incorporating a different alternative.

It is important to remember that in most cases we are predicting likely locations for case realizations, not necessary ones. If they fail to appear in the places predicted, it does not cast doubts on a theory. English allows considerable phrase juggling -- e.g. preposing prepositional phrases, fronting questioned phrases, etc. And, of course, not all predicted pre- and post-modifiers of a noun can occur to its immediate left or right. This must be remembered in considering how these local, frame-based predictions can be employed. Leftness and rightness constraints are implemented in SPEECHLIS as additional requests associated with proposals and monitors.

For example, consider Semantics processing a theory that a word match for "contain" was part of the original utterance. As mentioned earlier, "contain" heads the concept CONTAIN, whose other arguments are SAMPLE and CONSTITUENT. On both of these, monitors would be set to notice later instantiations of these concepts. Under the hypothesis that the clause is active, Semantics would include in the monitor set on the concept SAMPLE, the only possible active subject, that its instantiation be to the left of the match for "contain". In the monitor set on COMPONENT, the active object, we would indicate a preference for finding its instantiation to the right. This latter is only a preference because by question fronting, the object may turn up to the left, e.g. "What rare earth elements does each sample contain?". (Notice that regardless of where an instantiation of either SAMPLE or COMPONENT is found in the utterance, it will be noticed by the appropriate monitor. It is only how valuable the particular concept instantiation is to the theory setting the monitor that is affected by a positional preference.)

The process of checking the consistency of Syntax's and Semantics' hypotheses uses much the same information as that of making frame-based local predictions. As word matches are included in a theory, Semantics represents its hypotheses about their semantic structure in _case frame tokens_. These are instances of case frames which have been modified to show which word match or which other case frame token fills each instantiated case.

The two case frame tokens in Figure 3 represent a set of semantic hypotheses about how the word matches for "analyses", "ferrous" and "oxide" fit together. "Analyses" is the head (NP-HEAD) of a case frame token whose object case (NP-OBJ) is filled by another case frame token representing "ferrous oxide". Another way of showing this is in the tree format of Figure 4.

CASE FRAME TOKENS

```
[Cft #6
     ((( Realizes    Noun-Phrase ))
      (Np-Head (Analyses . 14 ) Nil Obl)
      (Np-Goal (Cft#5 . 1) (Of For ) Ellip)
      (Np-Loc (Mem . 7 ) ( In For Of On ) Ellip))]

[Cft #5
     ((( Realizes . Noun Phrase)
       ( Case of Cft #6 ))
      ( Np-Mod (Ferrous . 13) Nil Obl)
      ( Np-Head (Oxide . 5) Nil Obl )) ]
```

Figure 3.

SEMANTIC "DEEP STRUCTURE"

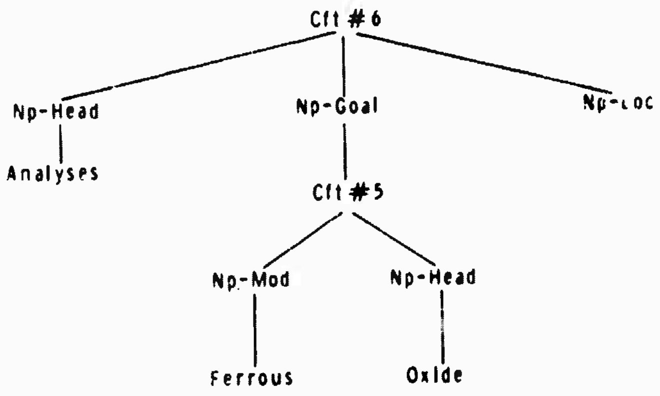


Figure 4.

Case frame tokens are used by Syntax to expedite the building of syntactic structures consistent with Semantic hypotheses and to evaluate the ones it has built with respect to fulfilling or violating those hypotheses. Syntactically, there are only a few ways of structuring the set of cases shown in Figure 3a. The head case must appear as the syntactic head and the object case must be realized either in a prepositional phrase or relative clause or as an adjectival modifier on the head. Thus, in Figure 5, syntactic structures (a) and (b) would confirm the semantic hypotheses in Figure 3, while (c), where "analyses" modifies "oxide", would not and would therefore receive a lower evaluation. Notice that the only difference between the terminal strings of (a) and (c) is the presence of the preposition "of". It takes only the presence of that small, acoustically ambiguous word to allow Syntax to build a structure consistent with Semantics' hypotheses. Knowing this, Syntax and Semantics should be able to work together to reconstruct and suggest to the word matcher these small function words which make all the difference for correct understanding.

SYNTACTIC STRUCTURES

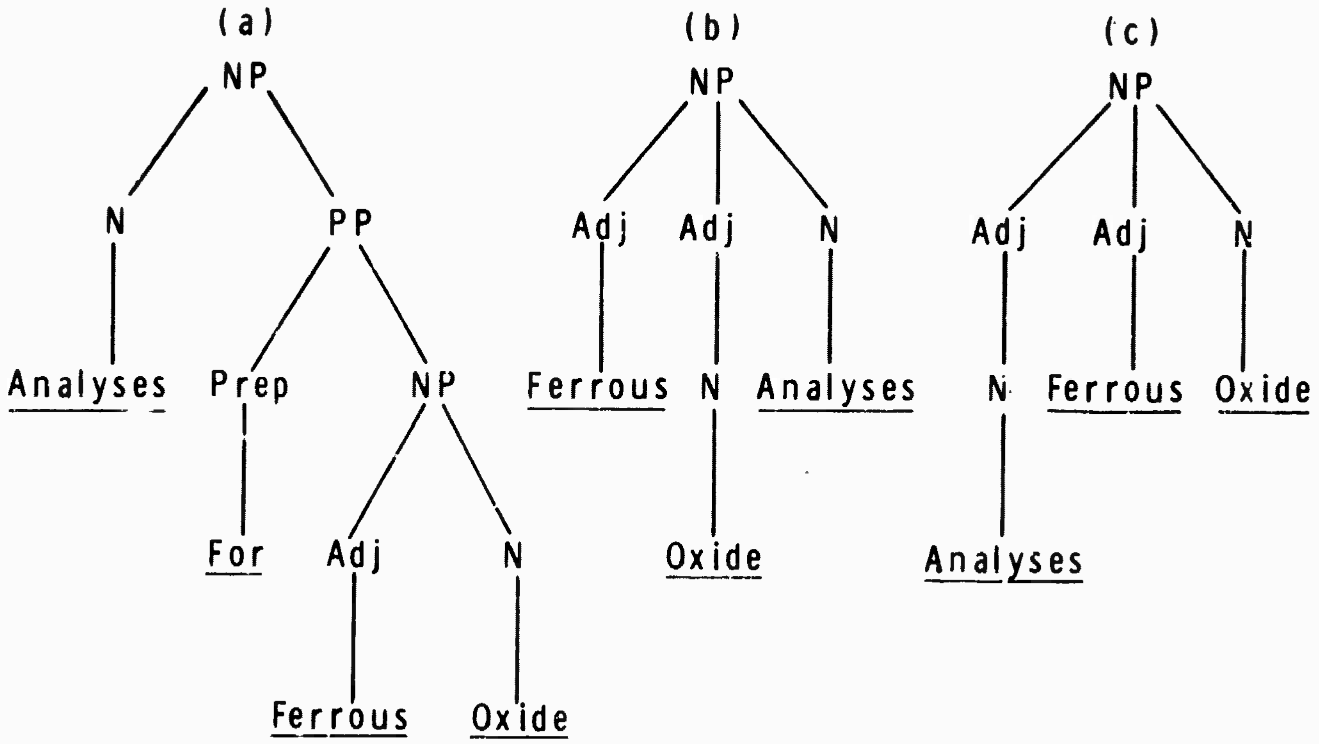


figure 5

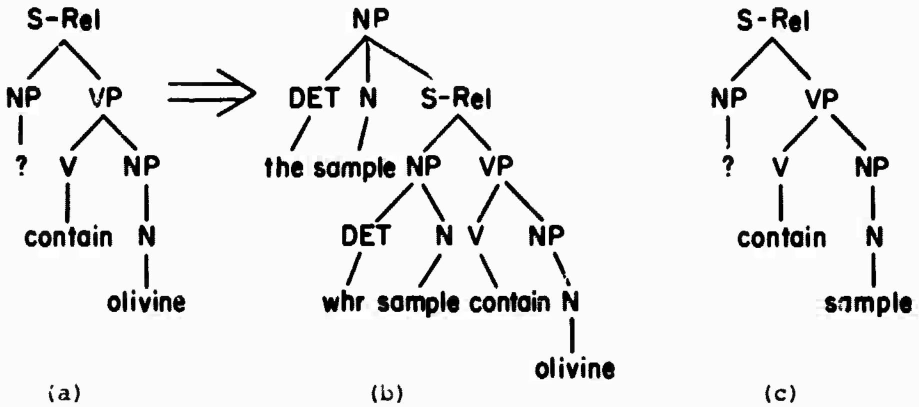
The point of the above discussion is that Syntax should not make choices randomly in places where Semantics has information that can be used to order them. This is implemented via Syntax's ability to ask questions of Semantics on the arcs of the Transition Network Grammar [3,44]. For example, noun/present-participle/noun strings may have the structure of a preposed relative clause like "the olivine containing sample" (i.e. "the sample which contains olivine") or a reduced relative clause like "the sample containing olivine". (It may be that prosodies help distinguish these two types of relative clauses in spoken utterances, but, as we suggested earlier, it may also be the case that this additional cue is not used if the phrase is already disambiguated by semantics or context.)

In parsing the string "the olivine containing sample", Syntax must choose whether the participle indicates a preposed relative clause or a reduced one. If preposed, "olivine containing" would have the structure shown in Figure 6a, with "olivine" as object and subject unknown. This is acceptable to Semantics, since olivine, a mineral, is a possible rock constituent and hence containable. "Sample" then becomes the head of the noun phrase and simultaneously the subject of the preposed relative clause, as shown in Figure 6b. This Semantics accepts. Were the word match one for "sulfur" instead of "sample", the final structure -- "the sulfur which contains olivine" -- would be semantically anomalous, and Semantics would advise Syntax to look for another possible parsing. On the other hand, "sample containing", with "sample" as object (Figure 6c), is semantically anomalous in the lunar rocks domain, so again Syntax would be advised to try again.

The olivine containing sample

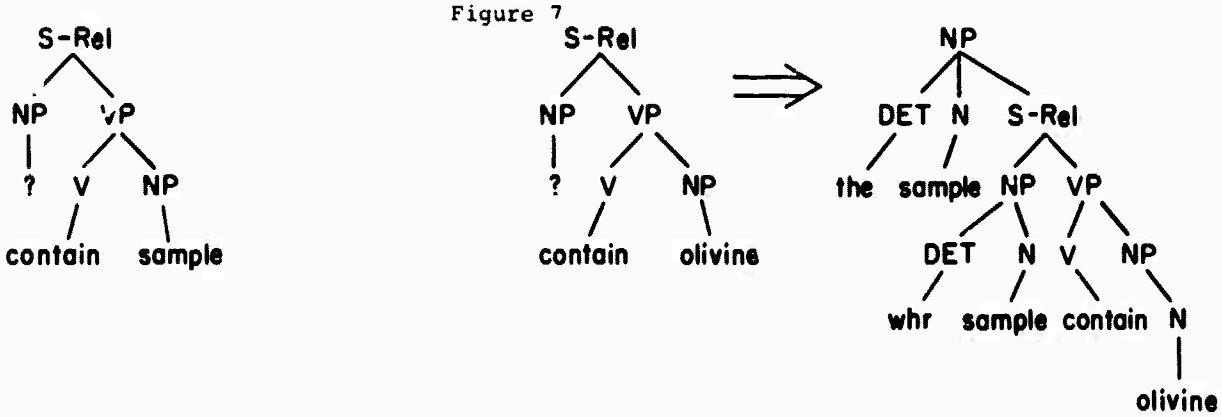
The sample containing olivine

Figure 6



(a) (b) (c)

Figure 7



(a) (b) (c)

As a normal relative clause, "the olivine containing sample" has the intermediate structure shown in Figure 7a, which is as bad as in 6c above. Only "The sample containing olivine" is reasonable as a normal reduced relative clause (Figures 7b and 7c). So Syntax's choice of parsing each string as a preposed or normal reduced relative clause will depend on its acceptability to Semantics.

D. Conclusions

Semantics is used in SPEECHLIS in several ways to aid the general speech understanding task. 1) It makes predictions local to a single utterance. 2) It collects sets of word matches which substantiate its hypotheses about the meaning of the utterance. 3) It checks the possible syntactic organizations of the word matches for confirmation or discrediting of those hypotheses. This it does using both a semantic network representing the concepts known in the domain and the words and multi-word names available for expressing them, and also case frames which give further information about their surface and syntactic realization.

The most important tasks we see before us now in regard to semantics and speech understanding are as follows:

(1) strengthening the bond between the syntactic and semantic components of the system, identifying specific useful points for their interaction and the types of information flow between them;

(2) formalizing (or at least clearly characterizing) the process of building a semantic network for speech understanding;

(3) writing a translation procedure from the syntactic and semantic representations of the utterance to one in the formal retrieval language, representing the intensional meaning of the utterance to the system. (We also plan to investigate whether this meaning representation can be usefully fed back into the system to help with hypothesis evaluation or to identify equivalent hypotheses.

(4) establishing an interface with the new user/task model currently under design and construction, in order to take into account pragmatic predictions about the content of an utterance as efficiently as possible.

We believe that semantic knowledge makes a very strong contribution to human speech understanding, and we will continue our work to make such knowledge available to automatic speech understanding as well.

VIII. PRAGMATICS - USER AND TASK MODEL

A. Introduction

The pragmatics component of a speech understanding system is a process which applies various facts about the speaker, the previous dialogue, and the domain of discourse to interpret utterances and respond appropriately. For example, the November 1973 BBN speech understanding system operates as a question answerer for the domain of lunar geology. Characteristics of the domain as well as the speaker's presumed perception of the system's function influence the way words are used. Thus, stative verbs like "contain" and "have" rarely appear in the past tense, while non-stative verbs like "find" and "analyze" rarely appear in the present. An intelligent system should be able to apply knowledge of this kind to predict, to evaluate interpretations, and to determine appropriate actions following an utterance.

Another example which arises in the lunar geology domain is based on the pragmatic principle that speakers tend to avoid using unnecessary words. For example, restrictive modifiers are normally used only when they perform a restricting function. For instance, in the phrase, "any people done chemical analyses" (from the sentence, "Have any people done chemical analyses on this sample?"), "people done" is not interpreted as a restrictive modifier on "chemical analyses" since, in this context chemical analyses are done only by people.

In the new travel budget management system we have recognized similar effects of pragmatics in simulated dialogues. For instance, following a supposition, a speaker typically asks a question. This question usually concerns future events and is related to the content of the supposition. Another example is that use of the vero "cancel" implies that the speaker believes that the object of "cancel" has been entered into the system data base.

In the November 1973 system, pragmatic tests are incorporated into the procedures for evaluating theories and events. These tests check such things as the likelihood of the hypothesized tense, aspect, voice and mood for a verb with respect to the context of lunar geology, e.g. the stative/non-stative tense distinction mentioned above. Other tests apply such facts as that in the lunar geology domain one is usually not concerned with the particular scientists who investigated the samples, but rather with the samples themselves. Thus verbs which allow agent deletion in the passive voice are usually expressed that way, rather than in the active voice. One says "Which new minerals were discovered in the lunar breccias?" and not "Which new minerals did the investigators discover in the lunar breccias?"

There is no doubt that pragmatics information can be helpful in certain cases. However, the _ad hoc_ introduction of pragmatics rules cannot be a general solution. For example, we might apply the rule about agent deletion to question the interpretation of an utterance as being, "Which new minerals did the investigators discover in the lunar breccias?", but it would be wrong to apply the rule to, "Have any people done chemical analyses on this sample?" In the latter case, the utterance is quite natural. The reason that our rule apparently fails is that in the context of lunar geology, "any people" is not a restrictive agent for "done." By not restricting, it serves as a null agent. This suggests a generalization of the agent deletion rule to something like, "Verbs which refer to actions done by people are usually expressed either as passives with agent deletion or as actives with a non-restrictive agent."

With the introduction of a second task domain, namely travel budget management, we are renewing emphasis on pragmatics. In the first place, it is important to generalize our techniques for applying pragmatics to speech understanding. In the second place, the new domain introduces some new elements, especially in the area of connected discourse. We are currently exploring the use of a user/task model to generalize and structure the pragmatics rules we have discovered. This model provides a focus on a central issue in pragmatics, the recognition of the speaker's purpose.

A person uses a speech system to accomplish some purpose, whether that be to obtain information, to gain assistance in planning and decision making, or to control some process. His purpose is reflected in both the vocabulary and syntax of the language and in the interpretations which are assigned to utterances. An at least implicit recognition of the purpose behind an utterance is necessary for complete speech understanding.

We have formulated a set of structures which can be used to represent the concept of intention in language use. These structures are based on analyses of simulated dialogues with the travel system, and on general considerations of what it means to communicate with a purpose. Discussion of the general considerations can be found in [8,9,38]. This section is primarily concerned with the more specific application of user and task knowledge to the travel budget speech understanding system.

Based on simulated dialogues with the travel system we have characterized several possible modes of interaction with the system and transitions between these modes. A session with the system then consists of a sequence of interaction modes. Modes are built out of other modes and intents. An intent is the smallest unit in our task model and represents the supposed purpose behind an utterance. An intent is, of course, somewhat sensitive to the mode one has hypothesized for the user. For example, if the user were to say, in edit mode, "Craig is also going to the ACL Meeting", one would say his intent was to make a permanent change to the data base. In query mode, however, (with a change in the intonation), one would say it was to get information from the data base.

In order to recognize intents and modes it is necessary to have a model of the speaker. The model includes such things as the speaker's presumed knowledge, his previous purposes, idiosyncratic pronunciation, vocabulary or syntax, and his role or position. Such a <u>user model</u> must be subject to change on the basis of interactions with the system.

The combination of a <u>task model</u>, expressed through modes and intents, and a user model can be a powerful aid to speech understanding. It can help first by providing expectations which structure the space of possibilities for utterances. For example, if the user says, "Suppose we cancel the upcoming Pittsburgh trip", the system can expect a question to follow, either immediately or after further suppositions. The question should be related to the suppositions and should refer to future possibilities. The fact that expectations are never certain does not invalidate their importance in suggesting possibilities. Thus the pragmatics component of the system can use the user/task model to indicate likely classes of morphemes (e.g., future tense indicators following a supposition), or structures for the next utterance.

Secondly, Pragmatics can use its user/task model to express preferences for certain readings over other ones. People certainly take into account what they suppose is the speaker's purpose when they hear an utterance. For example, when a gas station attendant says, "Fill´er up?", it is one´s understanding of his purpose which selects "Fill´er up?" over "Phil Rupp?".

Thirdly, Pragmatics can ensure that the actions of the system are appropriate to the goals of the user. If a user of the travel budget system were to say, "The cost of a flight to L.A. is two hundred dollars", he could be asking a question, attempting to insert new information into the system, or deliberately trying to change information in the data base. The system´s response might be either:

(1) No, it´s $250.
(2) My data base has $250 as the cost of a trip to L.A.
    Is that in error?
    or
(3) OK.

depending on what it discerns to be the user´s purpose.

In subsection B we consider a set of intents derived from examination of simulated uses of the travel budget system. Subsection C covers the organization of these intents into modes of interaction. Subsection D is a discussion of a sample dialogue with the system and the proposed actions of the pragmatics component using the user/task model. Subsection E is a discussion of implementation issues.

B. Intention in Speech

We can describe actions at many different levels. For example, the action -

> Susan said to Mary, "I hope you come tonight".

could be described as -

> Susan was facing Mary and uttered the sounds typically associated with the sentence, "I hope you come tonight".

On the other hand, a purpose oriented description might be -

> Susan urged Mary to come.

or in another context -

> Susan threatened Mary about coming.

The ability to generate purpose oriented descriptions for utterances is crucial for speech understanding because the speech act is always part of some plan directed towards a goal. General speech communication relies strongly on the ability of the communicators to maintain an awareness of the other's purposes. Underlying each utterance, then, is a purpose, or as we are calling it, an intent. In general an utterance can express any of several intents and an intent can be realized by many different utterances.

Before describing some intents we should sketch the context in which they are used. Imagine an observer of, or a participant in a dialogue. When he hears a sentence he immediately makes some interpretation. This interpretation may simply be that the speaker has chosen to inform his listeners that X, where X is some proposition. Whatever interpretation he makes, a rational observer commits himself to various beliefs. For example, the interpretation, "the speaker informed the hearer that X," commits him to the belief that the speaker believes X and that the hearer does not. Different beliefs correspond to different interpretations, e.g. "the speaker lied to the hearer that X" entails the belief that the speaker does not believe X. Beliefs of this kind are called <u>preconditions</u> since they refer to conditions prior to the utterance. There are also <u>outcome conditions</u> which refer to conditions after the utterance. For example, at least one sense of "inform" has the outcome condition that the hearer is aware of X. Both preconditions and outcome conditions are subject to later verification. If the observer later concludes that one or more of the conditions does not hold then he may change his interpretation of the utterance.

Each condition can be expressed as a formula consisting of a predicate with its arguments. Typically the predicates are such things as "believe" and "want", and the arguments (or <u>cases</u>) are such things as the speaker, the hearer, the time, and embedded propositions. (An embedded proposition might be the "X" in "the speaker believes X".) For further discussion of cases, see [7].

A full definition of an intent consists of its case structure, preconditions, outcome conditions, and a set of pointers to typical expressions of the intent in language. In the examples given here the case structures are all the same. There is an agent (the speaker), a recipient (the hearer), a time of utterance, and a proposition. We will symbolize these as A, R, T, and X respectively. Since each intent has the same case structure it will not be listed each time.

There are two preconditions applying to all intents which will not be listed explicitly in the examples to follow. First, the agent of the intent must intend to express that intent, i.e. he must be sincere. Regardless of the utterance, a given intent is realized only when the utterance is deliberately chosen (and not said as a joke. under duress, in a play, etc.). Second, the agent must believe that the recipient of the intent believes that the agent is sincere. If he does not then he has an obligation to supply additional information. Together these conditions imply what Searle [40] calls, "normal input/output conditions" for the speech act. Since one of the participants in the dialogues we are describing is SPEECHLIS itself, such notions as "sincerity" and "belief in sincerity" must be built into the user model and the system´s programmed interactions.

There is also a general outcome condition which says that if an observer (speaker, hearer, or third party) believes that an intent is expressed then he may compute any consequence of the preconditions or outcome conditions. For example, a since e "promise" has a precondition that the agent believes he can do the action promised. An observer of the promise might infer that the agent also believes that he has all the appropriate equipment and skills to do the action.

For the sake of readability the preconditions and outcome conditions for each intent are expressed in English. It is, however, possible to formalize these expressions (see [8,9]). The following are some of the intents found in travel budget management dialogues (square brackets indicate conditions believed by the agent).

ADD NEW STRUCTURED ITEM TO DATA BASE

(A "structured item" is a concept such as "trip" which is known to have specific components such as cost, travelers, destination, etc.)

Preconditions:

P1. A is user/R is system/X is a structured item
P2. [X is true]
P3. [X was not added before]
P4. [There is a standard set of questions based on the structure of X]
P5. [X is the kind of data item appropriate to the data base]

Outcome conditions

O1. X is added to data base
O2. R knows that A added X

Instances:

Add a trip for Bill to Berkeley.
Insert a new budget item.

ASK STANDARD QUESTION

(A "standard question" is one asked by the system to fill in a value for one of the components of a structured item, such as, "What is the cost of that trip?")

Preconditions:

P1. A is system/R is user/X is a question
P2 [R expects a question]
P3. [R will try to answer X]
P4. [R is adding a structured item to the data base]
P5. [X is relevant to this structured item]

Outcome conditions:

O1. A expects R to answer X

Instances:

What is the estimated cost for that trip?
What is the destination for that trip?
To what account should that trip be charged?

REPLY TO STANDARD QUESTION

Preconditions:

P1. A is user/R is system/X is data item
P2. [X is a direct answer to previous question of R]
P3. [A is adding a structured item to the data base]
P4. [X is consistent with previous replies for this structured item]

Outcome conditions:

01. X is added to data base
02. R knows that A added X

Instances:

Five hundred and fifty dollars.
L.A.
Account two-one-three-three-seven.

CONFIRM DATA ITEM

(The system should confirm that it has added new information to its data base.)

Preconditions:

P1. A is system/R is user/X is data item
P2. [X is comprehensible by A]
P3. [X is consistent with data base]
P4. [R expects confirmation of his last input]

Outcome conditions:

O1. R knows X has been added to data base*

Instances:

OK, cost is $350.

----

*A does not expect an answer or reply but will understand a negative statement indicating that the system has misunderstood. Otherwise the system may ask more questions (if any) or accept new interactions initiated by the user.

ASK AGAIN

(The system asks again when the user's response to a question is insufficient or inappropriate.)

Preconditions:

P1. A is system/R is user/X is question
P2. [R gave insufficient or inappropriate answer to a question of A]
P3. [The reason for the faulty answer was a misreading of the question]
P4. [R will recognize that X is the same question restated]
P5. [R will try to answer X]

Outcome conditions:

O1. A expects R to answer X

Instances:

I meant the total cost, air fare and taxis.

CONFIRM STRUCTURED ITEM

(The system should confirm that it has added a new structured item to its data base.)

Preconditions:

P1. A is system/R is user/X is structured item
P2. [X is complete]
P3. [R expects confirmation signal]

Outcome conditions:

O1. R knows X has been added to data base

Instances:

> OK, a new trip has been entered with the following structure: ...

EDIT

Preconc.tions:

P1. A is user/R is system/X is command to change an item in the data base
P2. [X refers to previously stored item]
P3. [effect of X is consistent with data base]

Outcome conditions:

O1. R app.ies X to data base if its effect is not inconsistent

Instances:

Change the registration fee to $75.
Add Bonnie to the list of people going to Chicago.

POINT OUT CONTRADICTION

Preconditions:

P1. A is system/R is user/X is data item
P2. [X is false with respect to other data]
P3. [R will try tc resolve contradiction]
P4. [R is not aware of conflict]

Outcome conditions:

O1. A expects R to resolve contradiction

Instances:

Is that figure correct?
Do you mean Pittsburgh? That destination was previously listed as Philadelphia.

REASSURE

(The user should respond in some way to the demonstration of a contradiction by the system. His response may be simply an assurance that the contradiction is unimportant or will be resolved later.)

Preconditions:

P1. A is user/R is system/X is data item
P2. [X is true]
P3. R has pointed out that X is inconsistent with other data

Outcome conditions:

O1. R accepts X

Instances:

That´s OK, enter the trip anyway.

STRONG EDIT

(The user expresses "strong edit" when he intends to make a change and expects that the system may find the change to be inconsistent.)

Preconditions:

P1. A is user/R is system/X is command to change an item in the data base
P2. X refers to a previously stored item
P3. [R believes X causes an inconsistency]

Outcome conditions:

01. R should apply X to data base
02. R should find that X is inconsistent with data base

Instances:

Change the registration fee, anyway.
I know it's inconsistent but go ahead and add Bonnie to the list of people going to Chicago.

Other intents have also been defined and are used in characterizing the modes of interaction. These include QUESTION, CLARIFY, QUERY, INFORM, PRESENT A SUPPOSITION, NAME SUPPOSITION, SUSPEND, TEST, and RESPOND.

In addition to the preconditions associated with each intent, there are assumptions which can be made about all communication within the travel budget world. These latter assumptions are essentially global presuppositions about utterances as opposed to the local presuppositions expressed as preconditions. One such global presupposition is that the travel budget system is helpful. While it may fail to assist the user in a particular case, its overall design is to help the user, not hinder, or ignore him. Another presupposition is that the user is bona fide, i.e. that he has the right to use the system and will not deliberately enter false information, nor attempt to foil the system. Certainly a system might not make these presuppositions and its actions would differ accordingly. However, the system's performance will benefit to the extent that global rules can be established.

C. Modes of Interaction

A direct consequence of the recognition of an utterance's intent is an expectation concerning the possible utterances which may reasonably follow. For example, if the travel budget system points out a contradiction in the data base then it can expect the user to respond with an utterance which realizes one of a few intents. He may rectify the data base, may assert that the contradiction is of no consequence, or may begin making tests of the data base to ascertain the reason for the discrepancy. Completely ignoring the system's comment is also a possibility, but it is not likely, especially in light of the global presuppositions that the system is trying to help and the user wants the system to be effective. An organization of intents into a larger structure expressing expectations we call a mode of interaction. Modes consist of (expectation) links between intents and (possibly) other modes. Thus the notion of "mode" is recursively defined.

Each mode is defined by a header and a body. The header determines whether or not the mode body is applicable in a given situation. In addition, it binds variables within the mode body to entities in the situation. The mode body is a graph in which the nodes are either intents or other modes, and the arcs are directed links between nodes, labelled by likelihood. In general, there is a small number (often one) of starting nodes in the mode body. The header requires that the preconditions for the starting mode intents be met. It may also impose other more general constraints, e.g. that the mode occurs only at the beginning of a session.

Currently, we have characterized the following modes of interaction:

(1) add - the user is attempting to add new information to the data base.

(2) conflict - the system has pointed out a contradiction between some statement or assumption made by the user and its own information. The user must then respond to it.

(3) edit - the user is attempting to change some information already in the data base.

(4) query - the user is attempting to get information from the system.

(5) question/clarify - the system does not understand the user's utterance and asks for clarification.

(6) supposition - the user is making hypothetical changes to the data base to see where they will lead.

(7) test - the user is attempting to ascertain that the system's knowledge about some past or future event conforms with his own.

These modes are presented in Figures 1 to 7. Abbreviations have been used to improve readability. The headers are omitted since in each case they simply check the preconditions on the starting modes. Variables for the intents are expressed implicitly by the shape of the box. An oval means the user is talking to the system; a rectangle means the system is talking to the user; a diamond means a recursive call to another mode. Likelihood ratings are also not given.

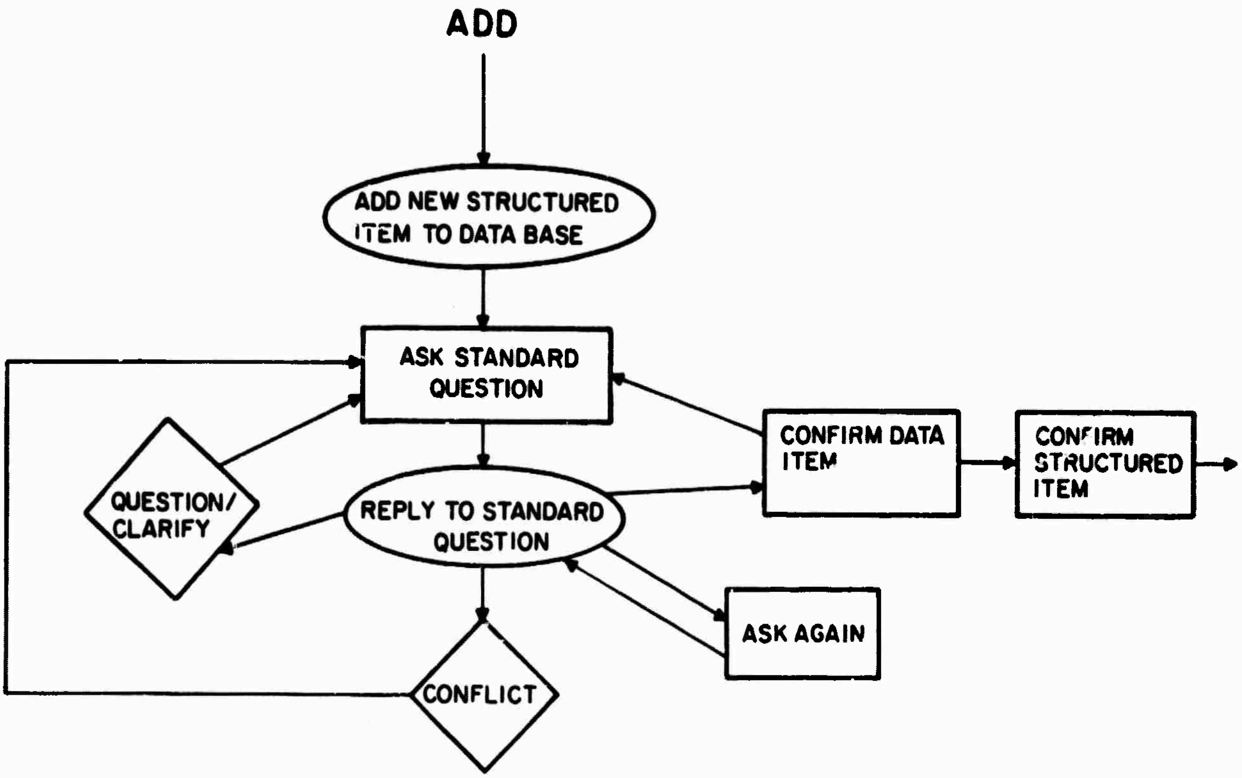


Figure 1.

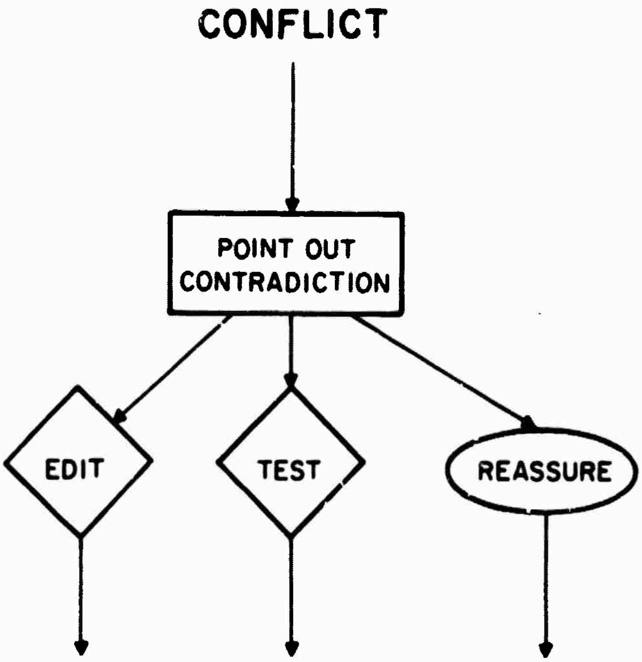


Figure 2.

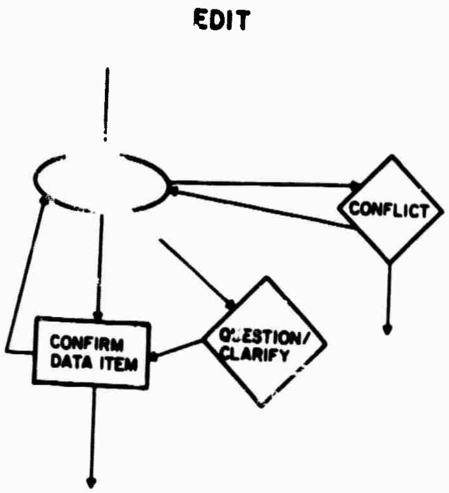


Figure 3.

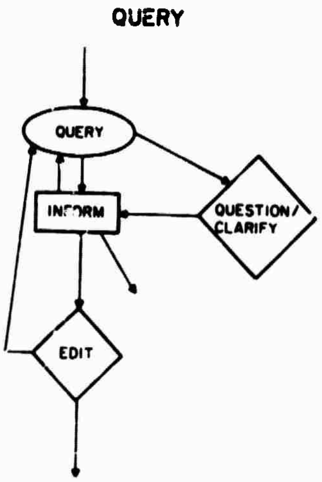


Figure 4.

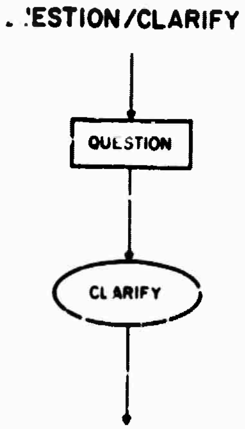


Figure 5.

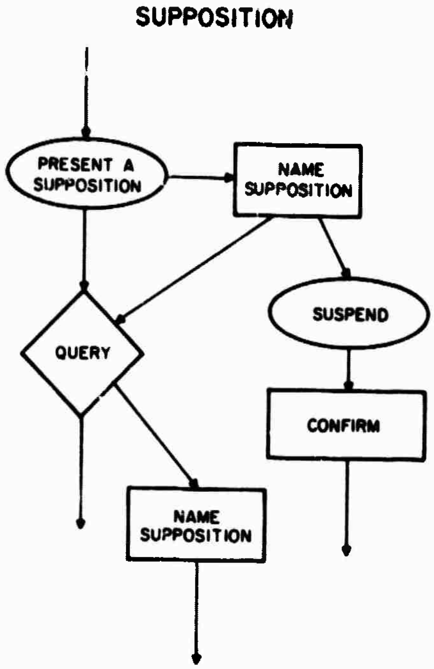


Figure 6.

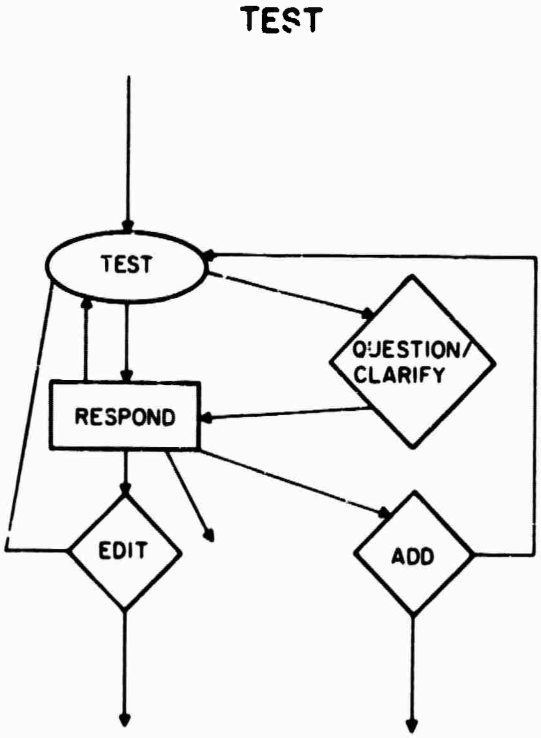


Figure 7.

While it would be too much to discuss each of the modes here, it may be helpful to describe one. A user enters *edit* mode (Figure 3) with the intention of changing some information in the data base. As a result of his utterance,

(1) the system may ask for clarification. That is, the mode may switch to question/clarify. Upon successful clarification, things proceed as in (3)) below.

(2) The system may point out a contradiction. For example, the user may have a mistaken assumption about what is actually in the data base. Here the mode switches to *conflict*.

(3) The system may make the requested change and confirm to the user that it has made it. At this point, the user may want to make another change, remaining in *edit* mode, or leave that mode for another one.

D. *Dialogue Analysis*

Perhaps the best way to understand how the user/task model we are building can be used in speech understanding is to analyze a simple dialogue. Consider the following interaction:

User: Give me a breakdown of the expenses to send one person to the London conference.

System:	Air fare (round trip)	$504
	Hotel,food,taxis (for one week)	$245
	Registration, miscellaneous	$ 50

	TOTAL:	$799

User: What would be the total budgeted amount for two people to London plus the untaken budget trips to other places?

System: Supposition #1
2 to London for a total of $1598

What do you mean by "budget trips"?

User: I meant "budgeted trips".

System: 2 to Pittsburgh for a total of $ 398
4.5 to Chicago for a total of $2200
1 to Washington for a total of $ 200

TOTAL: $4396

The pragmatics component uses its user/task model information about instances of intents to decide that the first sentence is a query. It would be considered an instance of a test if the system believed that the user knew the answer to the query. In that case the system might provide additional information such as the methods used in deriving the answer. After responding to the query, the system has a weak expectation for editing, since that more often follows the giving of information. However, it is also quite likely that a new mode will be entered.

Though the second user sentence looks like another query, the word "would" more strongly suggests a supposition. In fact the sentence is a supposition followed by a query. The system names the supposition, then enters the query mode. Since part of the user's utterance, "budget trips", was not understood, the system then goes into the question/clarify mode. Following its question the system has a very strong expectation for a clarification of "budget trips", e.g., a definition or a clearer pronunciation. The user then clarifies the misunderstanding, thus allowing the system to answer the original question.

E. Implementation Issues

The preceding sections have covered the use of a user/task model in speech understandirg. Such a model represents some of the knowledge needed by a general pragmatics component. In this section we discuss (1) what the pragmatics component should be able to do, (2) what implications its role has on communication with other components, (3) what implications its role has on the structure of the pragmatics component itself, and (4) what the current status is.

Pragmatics can perform several functions. For example, we might expect it to do any of the following:

(1) Following a portion of an utterance of the user it should express expectations regarding classes of morphemes to come. These expectations could go directly to Control or be filtered and refined via Syntax or Semantics.

(2) Given an interpretation of a word, phrase, or complete utterance, Pragmatics can be called to confirm or reject. For example, Syntax may need to insert an "is" or a "was" to complete a parsing. Pragmatics should be able to verify that one of these is likely for a given utterance. Semantics may suggest a reading for a noun group, in which case Pragmatics can confirm whether the construction is a plausible way of referring to some object and whether that object is likely to be referred to by the speaker in the current context.

(3) Given a complete utterance interpretation, Pragmatics can determine the intent of the utterance. On the basis of the intent it can decide what actions need to be performed and whether they are reasonable in terms of the user/task model, dialogue history, etc.

The functions of Pragmatics suggest that it needs to communicate with SPEECHLIS Control and, perhaps, directly with Syntax, Semantics, and the factual data base. We are currently exploring the establishment of these communications channels.

Pragmatics itself requires a control structure which allows access to varied sets of data. A preliminary design is shown in Figure 8. It is essentially a single coordinating process called the *Pragmatics Control* plus a set of *knowledge sources* and a *context representation*. The knowledge sources include the definitions of intents and modes. The context representation consists of the *mode status* (the current mode and state within the mode), a representation of the *facts* of the dialogue (i.e. the system's knowledge), the system's representation of the user's facts (the *user's knowledge*), and a *dialogue history*, which contains such things as information about likely ways of referring to objects. This latter element is especially important for problems of anaphora and ellipsis.

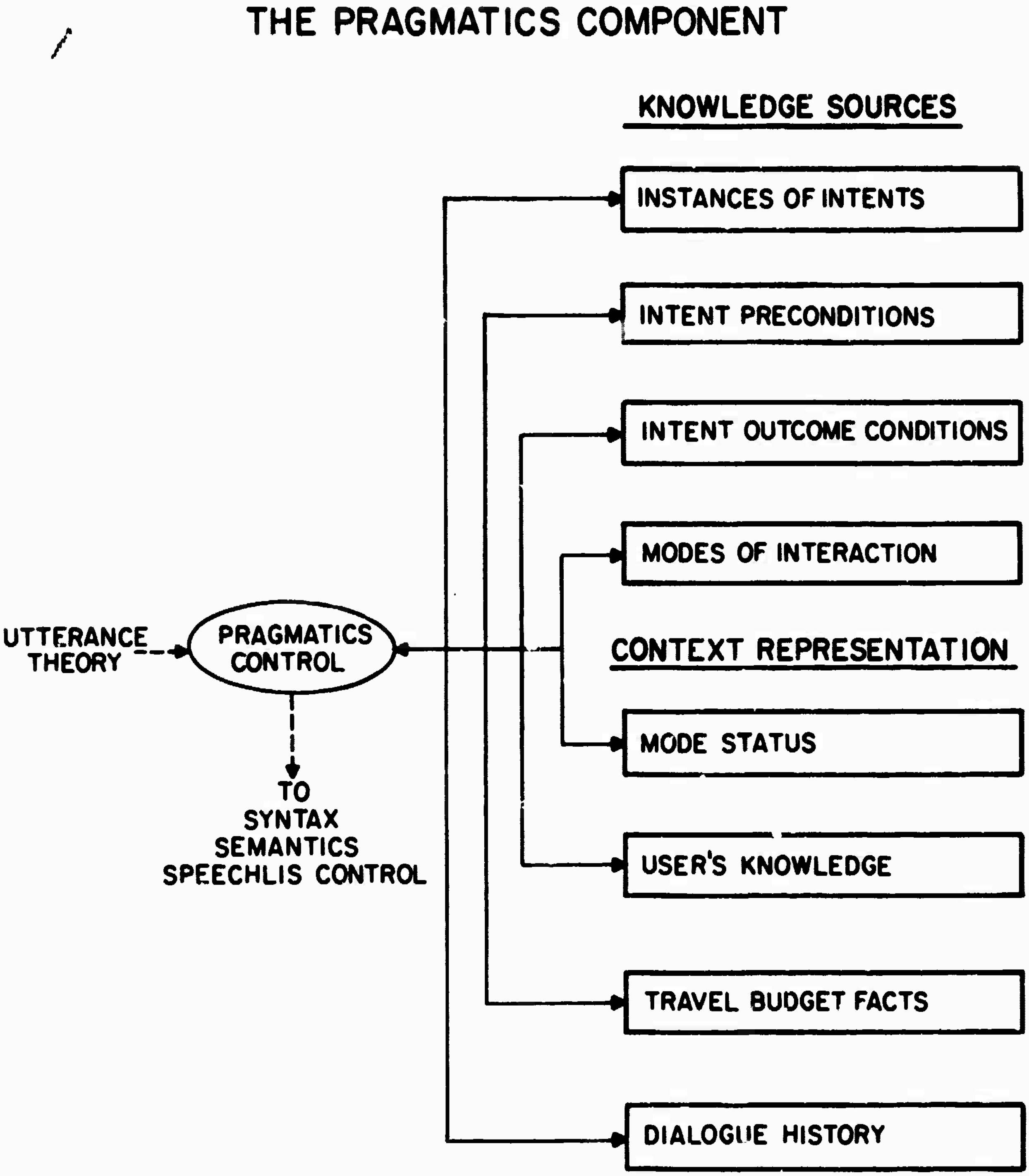

THE PRAGMATICS COMPONENT
KNOWLEDGE SOURCES
INSTANCES OF INTENTS
INTENT PRECONDITIONS
INTENT OUTCOME CONDITIONS
MODES OF INTERACTION
UTTERANCE THEORY
PRAGMATICS CONTROL
CONTEXT REPRESENTATION
MODE STATUS
TO SYNTAX SEMANTICS SPEECHLIS CONTROL
USER'S KNOWLEDGE
TRAVEL BUDGET FACTS
DIALOGUE HISTORY

Figure 8.

The pragmatics control can be called whenever an interpretation of an utterance (or portion of an utterance) is to be evaluated or responded to by the system. Pragmatics Control first looks at instances of intents. This is a knowledge source which defines a mapping of words and phrases into intents. Using simple pattern matching rules various intents may be suggested. These suggestions can be supported or rejected by consideration of the mode status. Pragmatics Control looks there to determine what, if any, intents can be expected in the current context. Given this filtering of the possible intents, Pragmatics Control can then begin to select the probable intent using the intent preconditions. These may require significant computations on both the travel budget facts and the user´s knowledge. Once an intent is selected, Pragmatics Control processes its outcome conditions, changing the context representation as needed. The output of the pragmatics control can be a message to SPEECHLIS Control, such as a verification of the utterance interpretation, a request for actions on the data base, or notes to Semantics or Syntax concerning the subsequent utterance, i.e. words or classes of words to look for.

The current work on pragmatics within the speech system represents a compromise between the ideal of a general pragmatics component which truly understands human motivations and the reality of a working system. Further development of the user/task mode outlined above will provide a framework in which otherwise ad hoc pragmatics principles can be implemented.

IX. CONCLUSION

The system described in this report is an intermediate step in the development of an evolving system. It represents the current state of our attempt to embody in computer algorithms thosc techniques which we think will be required to solve difficult problems of speech understanding. There remain many problems for which we do not yet have solutions, many areas in which we are not satisfied with our current techniques, and many planned techniques which have not yet been implemented and put to the test. In this chapter, we would like to illustrate some of the kinds of things we have learned from the project so far, and some of the directions for the future.

As mentioned in the introduction, we have learned a great deal from early incremental simulations of a total system. In particular, the different modes for handling small function words and content words became apparent as a result of such simulations as well as the observation that for handling many cases of garbled or misanalyzed words it is important to be able to skip over them to obtain an analysis of the rest of the sentence and to use this information to try to infer the missed word. Both of these observations result in an overall control mechanism that is more cumbersome than a straightforward left-to-right, top-down parsing with a strongly constraining grammar, but we are convinced that some such mechanism is essential for the more difficult cases.

Certain other things were known qualitatively at the outset of the project, but the depth and detail of our understanding of the problems has increased as a result of simulations and experience gained from running the November 1973 system. For example, we have been aware from the outset that coping with the various combinatorial problems would be one of the more difficult aspects of the speech understanding problem, but the appreciation of such techniques as the clustering of "fuzzy" word matches and semantically equivalent word matches resulted from observations of system behavior in the partially simulated, partially implemented mode. The effectiveness of including differential deletion likelihoods and duration checks based on stress markings for the phonemes within a word match have been proven by observing the success of the lexical retrieval component with and without such techniques, and our reading experiments suggest that there is much additional benefit to be derived from sophisticated word matching techniques. It is towards this end that we are attempting to construct an analysis-by-synthesis type word verification component based on Klatt's synthesis-by-rule program to verify words at the parametric level.

Similarly, we have known from the outset that the level of detailed knowledge that must be incorporated in the acoustic/phonetic analysis component was much greater than that which could be included in the November 1973 system. Spectrogram reading and parameter reading experiments have sharpened our knowledge of specific acoustic/phonetic facts that need to be incorporated, and critical analysis of the acoustic/phonetic analyzer in the November 1973 system has helped us tc design techniques for incorporating this information effectively in our new acoustic/phonetic component. During the past year especially, we have collected and codified a substantial set of acoustic/phonetic and phonological rules, which will be incorporated into the system in various ways. An experimental system for performing statistical evaluations of quantitative and algorithmic embodiments of acoustic/phonetic facts has been constructed. We have high hopes for significant improvements in capability during the coming years.

A. Difficult Problems

Some of the problems that we are dealing with are instances of known difficult problems in the fields of linguistics, computational linguistics, and artificial intelligence. For example, the use of semantic information to guide parsing, the use of pragmatic and factual knowledge and inferences from this knowledge to determine the intent of an utterance, and the characterization and use of different degrees of grammaticality or likelihood of syntactic construction are all difficult problems that have been studied in other fields for some time and not solved (although there are various partial solutions or attempts at solutions). Thus it is not the case that we are merely applying solutions from other fields to problems in speech understanding, but we must in fact break new ground in some of those fields. For the most part, we have attempted to structure our speech understanding tasks not to require radical breakthroughs in these other fields, and are attempting where possible to restrict ourselves to problems where existing artificial intelligence and language analysis techniques can be effective. However, these techniques cannot simply be carried over to the speech applications without modification. For example, the current (text oriented) techniques for using semantic and pragmatic information to aid parsing that have been developed in the field of computational linguistics have disadvantages when carried over to speech understanding. A considerable portion of our work so far and for the remainder of the project must go into discovering, evaluating, and modifying techniques for the effective interaction among the syntactic, semantic, and pragmatics components during an analysis of a speech utterance. We have learned a lot from our experience so far, and we are continuing to strenghthen the interactions between these components, but there remains much to be done in this area, and much is likely to remain beyond the end of the current 5-year project.

Another known difficult problem is the interaction of the prosodics of speech -- the intonation contour durations, hesitations, rhythm, etc. --- with the syntactic structure and intended effect of an utterance in context. This problem has been studied for some time by linguistics in subjective terms, but there have been few instrumented studies in terms of quantitative, measurable characteristics of the utterance. Recent work, largely stimulated by the current ARPA speech project, has begun to remedy the lack of such knowledge, and hopefully some of it will be useable in speech understanding systems in the near future. However, it is clear that the need for more study in this area will extend far beyond the current 5-year program. In the BBN speech project, resource limitations prevent us from attempting a major study of speech prosodics on our own, but we are cooperating with the prosodics groups at UNIVAC and at the University of California at Berkeley, in hopes of gaining prosodic techniques which can help reject erroneous interpretations of speech signals or choose between competing ones. We have encountered several examples where such information would have been helpful, and we have a rudimentary understanding of where they could fit into the overall control strategy. However, we do not yet have mechanical prosodic cue detectors which we can incorporate into our incremental simulations to refine these ideas.

B. A Vision of the Five-Year Mark

In summary, we have come a long way toward developing an insight into speech understanding problems and developing techniques for dealing with them since the inception of the speech project, and we anticipate making considerable additional progress during the coming years. Our objective at the 5-year mark is to have developed a technology for speech understanding which approximates that outlined in the ARPA speech study group report. Furthermore we hope to understand it well enough to say what problems are beyond the capabilities of that technology. At that time we will have a computer implementation which illustrates the technology and demonstrates a level of achievement. It is likely that there will be practical speech understanding tasks which can be handled with this level of technology and one of our goals is to be able to evaluate such applications for potential practical development. However, it is clear that even if we are totally successful in our objectives for the 5-year mark, there will remain significant speech understanding problems which have not been faced and which will require further research before they can be solved. Our hopes for the 5-year system are that in addition to suggesting practical applications of this technology it will also demonstrate the feasibility and potential payoff of continued research on the difficult problems.

REFERENCES

[1] Aho, A.V. and Ullman, J.D., The Theory of Parsing, Translation, and Compiling, Prentice-Hall Inc., Englewood Cliffs, New Jersey (1972).

[2] Barnett, J.A., "A Phonological Rule Compiler," Proc. IEEE Symposium on Speech Recognition, CMU, pp. 188-192 (April 1974).

[3] Bates, M., "The Use of Syntax in a Speech Understanding System," Proc. IEEE Symposium Speech Recognition, CMU (April 1974).

[4] Bobrow, D.G. and J.B. Fraser, "A Phonological Rule Tester," CACM 11, pp. 766-772 (1968).

[5] Bolinger, D., "Accent is Predictable (if you're a Mind Reader)", Language 48(3), pp. 633-644 (1972).

[6] Broad, D.J., "Formants in Automatic Speech Recognition," Int. J. Man-Machine Studies, Vol. 4, pp. 411-424 (July 1972).

[7] Bruce, B., "Belief Systems and Language Understanding," BBN Report No. 2973, Bolt Beranek & Newman, Camb., Ma. (1974).

[8] Bruce, B., "Case Systems for Natural Language." Rutgers Computer Science Dept. Report CBM-TR-31 (1974) .

[9] Bruce, B. and C.F. Schmidt, "Episode Understanding and Belief Guided Parsing", Computer Science Department, Rutgers, NIH Report, CBM-TR-32 (1974) .

[10] Carbonell, J. and A.M. Collins, "Natural Semantics in Artificial Intelligence," Proc. 3rd IJCAI, Stanford, Ca. (August 1973).

[11] Colarusso, J., "Phonological Rules for Continuous Speech, SUR Note No. 133, NIC No. 30487 (1974).

[12] Collins, A.M. and E. Warnock, "Semantic Networks," BBN Report No. 2833, Bolt Beranek and Newman Inc., Cambridge, Mass. (1974).

[13] Denes, P. and E. Pinson, The Speech Chain, Bell Telephone Laboratories, Murray Hill, New Jersey (1963).

[14] Earley, J., "An Efficient Context-Free Parsing Algorithm," CACM, Vol. 13, No. 2, pp. 94-102 (February 1970).

[15] Fant, C.G.M., "Descriptive Analysis of the Acoustic Aspects of Speech," LOGOS, Vol. 5, No. 1, pp. 3-17 (April 1962).

[16] Fillmore, C., "The Case for Case", in Bach and Harms, Universals in Linguistic Theory, pp. 1-90 (1968).

[17] Fromkin, V., "The Non-anomalous Nature of Anomalous Utterances", Language, 47(1), pp. 27-53 (1971).

[18] Hewitt, C., "Description and Theoretical Analysis (using Schemas) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot," Ph.D. Thesis, M.I.T. (February 1971).

[19] Kaplan, R.M., "Augmented Transition Network Grammars as Psychological Models of Sentence Comprehension", Proceedings, 2nd IJCAI, London (1971).

[20] Klatt, D.H., "Word Verification in a Speech Understanding System", invited paper, IEEE Symposium on Speech Recognition, Carnegie-Mellon University, April 15-19 (1974), in Speech Recognition: invited papers presented at the IEEE symposium, R. Reddy (ed.), Academic Press, (in press).

[21] Klatt, D.H. and K.N. Stevens, "Strategies for Recognition of Spoken Sentences from Visual Examination of Spectrograms," BBN Report No. 2154, Bolt Beranek and Newman Inc., Cambridge, Mass. (1971).

[22] Klatt, D.H. and K.N. Stevens, "Sentence Recognition from Visual Examination of Spectrograms and Machine-Aided Lexical Searching," Conference Record, 1972 Conference on Speech Communication and Processing, Newton, Mass. (April 1972).

[23] Klatt, D.H. and K.N. Stevens, "On the Automatic Recognition of Continuous Speech: Implications from a Spectrogram-Reading Experiment," IEEE Trans. on Audio and Electroacoustics, AU-21, No. 3, pp. 210-217 (June 1973).

[24] Makhoul, J., "Spectral Analysis of Speech by Linear Prediction," IEEE Trans. on Audio and Electroacoustics, AU-21, No. 3, pp. 140-148 (June 1973).

[25] Makhoul, J., "Selective Linear Prediction and Analysis-by-Synthesis in Speech Analysis," presented at the November 1973 ASA meeting in Los Angeles, also BBN Report No. 2578, Bolt Beranek and Newman Inc., Cambridge, Ma. (1974).

[26] Makhoul, J., "Linear Prediction in Automatic Speech Recognition," invited paper, IEEE Symposium on Speech

Recognition, Carnegie-Mellon University, April 15-19, 1974, in *Speech Recognition: invited papers presented at the IEEE symposium*, D.R. Reddy (ed.), Academic Press (in press).

[27] Makhoul, J., "Linear Prediction: A Tutorial Review," IEEE Proceedings special issue on digital signal processing, (to appear) (April 1975).

[28] Makhoul, J. and J. Wolf, "Linear Prediction and the Spectral Analysis of Speech," BBN Report No. 2304 (AD749-066), Bolt Beranek and Newman Inc., Cambridge, Mass. (August 1972).

[29] Makhoul, J. and J. Wolf, "The Use of a Two-Pole Linear Prediction Model in Speech Recognition," BBN Report No. 2537, Bolt Beranek and Newman Inc., Cambridge, Ma., also presented at the April 1973 ASA meeting in Boston, (1973).

[30] Marcus, M., "Wait-and-See Strategies for Parsing Natural Language", MIT Artificial Intelligence Laboratory Working Paper 75 (1974).

[31] Nash-Webber, B., "Semantic Support for a Speech Understanding System," Proc. IEEE Symposium on Speech Recognition, CMU (April 1974).

[32] Nash-Webber, B., "The Role of Semantics in Automatic Speech Understanding," in *Representation and Understanding: Studies in Cognitive Science*, D.Bobrow and A. Collins (eds.), Academic Press (in press).

[33] Newell, A. et al., *Speech-Understanding Systems: Final Report of a Study Group*, North-Holland/American Elsevier (1973).

[34] Oshika, B.T., "The Role of Phonological Rules in Speech Understanding Research, Proceedings IEEE Symposium on Speech Recognition, pp. 204-207 (April 1974).

[35] Riesbeck, C.K., "Computational Understanding: Analysis of sentences and context", Ph.D. Thesis, Stanford University (1974). (Also reprinted in part in Schank, R. (ed.), *Conceptual Information Processing*, North-Holland (1974).)

[36] Rovner, P., B. Nash-Webber and W. Woods, "Control Concepts in a Speech Understanding System," BBN Report No. 2703, Bolt Beranek and Newman Inc., Cambridge, Ma. (also Proc. IEEE Symposium on Speech Recognition, CMU) (1974).

[37] Rovner, P., J. Makhoul, J. Wolf and J. Colarusso, "Where the Words Are: Lexical Retrieval in a Speech Understanding System," Proc. IEEE Symposium on Speech Recognition, CMU

(April 1974).

[38] Schmidt, C.F., "Recognizing Plans and Purposes", Computer Science Department, Rutgers, NIH Report, CBM-TR-34 (1974).

[39] Schwartz, R. and J. Makhoul, "Where the Phonemes Are: Dealing with Ambiguity in Acoustic-Phonetic Recognition," Proc. IEEE Symposium on Speech Recognition, CMU (April 1974).

[40] Searle, J.R., Speech Acts: An Essay in the Philosophy of Language, Cambridge, England, Cambridge University Press, (1969).

[41] Shapiro, S., A Data Structure for Semantic Information Processing". Unpublished Ph.D. dissertation, University of Wisconsin, Madison, Wisconsin (1971).

[42] Wanner, E., "Do We Understand Sentences from the Outside-In or from the Inside-Out?" Daedalus, pp. 163-183 (Summer 1973).

[43] Winograd, T., "PROGRAMMER: A language for Writing grammars", MIT Artificial Intelligence Laboratory Memo No. 181 (1969).

[44] Woods, W.A., "Transition Network Grammars for Natural Language Analysis," Communications of the ACM, Vol. 13, No. 10, pp. 591-602 (October 1970).

[45] Woods, W.A., "An Experimental Parsing System for Transition Network Grammars," in R. Rustin (ed.) Natural Language Processing, Algorithmics Press, New York, pp. 111-154 (1973).

[46] Woods, W.A., "Progress in natural language understanding -- An application to lunar geology," AFIPS Proceedings, 1973 National Computer Conference and Exposition (1973).

[47] Woods, W.A., "Motivation and Overview of BBN SPEECHLIS: An Experimental Prototype for Speech Understanding Research," Proc. IEEE Symposium on Speech Recognition, CMU (April 1974).

[48] Woods, W.A., M. Bates, J. Colarusso, J. Makhoul, B. Nash-Webber, P. Rovner. R. Schwartz and J. Wolf, "Speech Understanding Research: Collected Papers 1973-74," BBN Report No. 2856, Bolt Beranek and Newman Inc., Cambridge, Ma. (1974).

[49] Woods, W.A., "Syntax, Semantics, and Speech," invited paper, IEEE Symposium on Speech Recognition,

Carnegie-Mellon University, April 15-19, 1974, in _Speech Recognition: invited papers presented at the IEEE symposium_, D.R. Reddy (ed.), Academic Press (in press).

[50] Woods, W.A. and J. Makhoul, "Mechanical Inference Problems in Continuous Speech Understanding," _Proceedings of the Third International Joint Conference on Artificial Intelligence_, pp. 200-207 (August 1973). (Reprinted in _Artificial Intelligence_, Vol. 5, No. 1, pp. 73-91 (Spring 1974).

[51] Woods, W.A., R.M. Kaplan and B. Nash-Webber, "The Lunar Sciences Natural Language Information System: Final Report," BBN Report No. 2378, Bolt Beranek and Newman Inc., Cambridge, Ma. (June 1972).

APPENDIX A

I. HARDWARE

We have specified and procured several items of equipment primarily in support of the speech understanding project, but also for the network speech compression project, described elsewhere, principally for graphics displays and analog signal handling and digitization.

A. Graphics

An IMLAC PDS-1 graphic display computer was acquired in 1971. This is a 16-bit minicomputer with a separate display processor, which drives a 14 inch CRT display. Our machine has 16K of memory, a tablet, mouse and keyset, hard-copy display, and a 9600 Baud asynchronous connection to three of the PDP-10's. We have constructed four 16-bit toggle registers and six knobs to give us additional operator interaction facilities and a high speed parallel interface, which will give us much faster communication with TENEX. We have developed two major systems programs for the IMLAC: TSIM, a simple monitor which allows an applications display program to run in the IMLAC and interact with a TENEX process, and IMSYS, a general purpose graphics system whose display can be manipulated via procedure calls from LISP, FORTRAN, or BCPL processes running in TENEX. This graphics facility has proved indispensible, particularly for the work in signal processing and acoustic-phonetic recognition, and it is also used by other projects which use BBN's computers.

B. Analog Signal Handling and Digitization

Processing speech signals requires the ability to convert them back and forth between digital and analog form. Our initial work on speech understanding used digitizations done outside BBN. Later we were able to use the A/D converter at Lincoln Laboratories via a program quality telephone line and the ARPANET. However, these were just stopgap measures until a Real Time Interface for the System-B TENEX could be built. These required special changes to be made to the system-B monitor in order to operate a real-time process such as A/D and D/A conversion at the very high bit rates required by speech. Unfortunately, while these changes have enabled us to use the RTI at a 10 kHz sampling rate, we have not been able to use it at the desired 20 kHz rate.

This need, and the need for more efficient signal processing computation, have led to the design, in close consultation with the other ARPA speech understanding and speech compression projects, of a system built around a PDP11/40 and an SPS-41 signal processing computer. This system, all the pieces of which have not yet been delivered, will include dual 12-bit A/D and D/A converters, a 30 million word disk, 56 K of core memory for the PDP11 plus 8K of semiconductor memory shared between the two processors, and the prototype "ARPA standard" PDP11-ARPANET interface.

This PDP11/SPS41 system was designed to be similar in many respects to the systems being assembled by the other speech understanding and speech compression contractors. Accordingly we plan to cooperate quite closely in software development for these systems. Indeed, this is already happening in the case of several pieces of SPS-41 support software and signal processing program modules.

APPENDIX B

PUBLICATIONS

Makhoul, J. and J. Wolf, "Linear Prediction and the Spectral Analysis of Speech," BBN Report No. 2304 (1972).

Abstract

This report gives a detailed treatment of the use of linear prediction in speech analysis. New concepts are developed and more familiar concepts are seen in a new way. The Covariance and Autocorrelation methods are derived in the time and frequency domains. Both methods are shown to be derivable from a more general concept, that of generalized analysis-by-synthesis, where a nonstationary two-dimensional spectrum is approximated by another model spectrum. Linear prediction analysis is a special case where the model spectrum is all-pole. Also, under the assumption of stationarity the general Covariance method reduces to the Autocorrelation method. The normalized error is defined. Its relation to the cepstral zero quefrency, its usefulness as a voicing detector and as a determiner of the optimum number of predictor coefficients are discussed. The application of linear prediction to pitch extraction and formant analysis is carefully examined. Specific issues discussed include the adequacy of an all-pole model for formant extraction, pitch-synchronous and pitch-asynchronous analysis, windowing, preemphasis, and formant extraction by peak picking.

Makhoul, J. and J. Wolf, "The Use of a Two-Pole Linear Prediction Model in Speech Recognition," BBN Report No. 2537 (1973).

Abstract

In speech recognition applications, it is often desirable to make a gross characterization of the shape of the spectrum of a particular sound. The autocorrelation method of linear prediction analysis leads to an all-pole approximation to the signal spectrum. Hence an LPC analysis using two poles produces one possible gross characterization. The two poles are computed as the roots of a quadratic equation whose coefficients are the linear prediction parameters, which are simple functions of the autocorrelation coefficients R , R , and R . The poles are either both real or form a conjugate pair in the z plane. This fact, together with the exact positions of the poles, is particularly useful in describing certain gross characteristics

of the spectrum. The spectral dynamic range of the two-pole spectrum and the normalized minimum error are suggested as more suitable substitutes for the two-pole bandwidths in interpreting the information supplied by the model for the purpose of spectral characterization.

Woods, W. and J. Makhoul, "Mechanical Inference Problems in Continuous Speech Understanding," BBN Report No. 2565 (1973).

Abstract

Experiments by Klatt and Stevens at MIT indicate that the process of deciphering the content of spoken sentences requires a close interaction between the acoustic/phonetic analysis of the speech signal and higher level linguistic knowledge of the listener. This paper describes a technique of "incremental simulation", which is being used to discover the different roles of syntactic, semantic, pragmatic, and lexical information in this process and to evolve effective strategies for applying these different types of knowledge in a computer system for understanding continuous speech. Two examples illustrate the situations in which the different sources of information make their contributions and the types of probabilistic, plausible inference techniques which are required to take advantage of them.

Rovner, P., B. Nash-Webber and W. Woods, "Control Concepts in a Speech Understanding System," BBN Report No. 2703 (1973), (also Proc. IEEE Symposium on Speech Recognition, CMU) (1974).

Abstract

Automatic speech understanding must accomodate the fact that an entirely accurate and precise acoustic transcription of speech is unattainable. By applying knowledge about the phonology, syntax, and semantics of a language and the constraints imposed by a task domain, much of the ambiguity in an attainable transcription can be resolved. This paper deals with how to control the application of such knowledge. A control framework is presented in which hypotheses about the meaning of an utterance are automatically formed and evaluated to arrive at an acceptable interpretation of the utterance. This design is currently undergoing computer implementation as a part of the BBN Speech Understanding System (SPEECHLIS).

Makhoul, J., "Selective Linear Prediction and Analysis-by-Synthesis in Speech Analysis," BBN Report No. 2578 (1974).

Abstract

Linear prediction is presented as a spectral modeling technique in which the signal spectrum is modeled by an all-pole spectrum. The method allows for arbitrary spectral shaping in the frequency domain, and for modeling of continuous as well as discrete spectra (such as filter bank spectra). In addition, using the method of selective linear prediction, all-pole modeling is applied to selected portions of the spectrum, with applications to speech recognition and speech compression. Linear prediction is compared with traditional analysis-by-synthesis techniques for spectral modeling. It is found that linear prediction offers computational advantages over analysis-by-synthesis, as well as better modeling properties if the variations of the signal spectrum from the desired model are large. For relatively smooth spectra and for filter bank spectra, analysis-by-synthesis is judged to give better results. Finally, a suboptimal solution to the problem of all-zero modeling using linear prediction is given.

Makhoul, J. and R. Viswanathan, "Quantization Properties of Transmission Parameters in Linear Predictive Systems," BBN Report No. 2800 (1974).

Abstract

Several alternate sets of parameters that represent the linear predictor are investigated as transmission parameters for linear predictive speech compression systems. Although each of these sets provides equivalent information about the linear predictor, their properties under quantization are different. The results of a comparative study of the various parameter sets are reported. Specifically it is concluded that the reflection coefficients are the best set for use as transmission parameters. A more detailed investigation of the quantization properties of the reflection coefficients is then carried out using a spectral sensitivity measure. A method of optimally quantizing the reflection coefficients is also derived. Using this method it is demonstrated that logarithms of the ratios of the familiar area functions possess approximately optimal quantization properties. Also, a solution to the problem of bit allocation among the various parameters is presented, based on the sensitivity measure.

The use of another spectral sensitivity measure renders logarithms of the ratios of normalized errors associated with linear predictors of successive orders as the optimal

quantization parameters. Informal listening tests indicate that the use of log area ratios for quantization leads to better synthesis than the use of log error ratios.

Woods, W. et al., "Speech Understanding Research: Collected Papers 1973-74," BBN Report No. 2856 (1974).

Abstract

This report consists of a collection of papers describing the BBN Speech Understanding system, a research prototype computer system designed to understand and respond appropriately to instructions, commands, and questions expressed in ordinary continuous speech. This system attempts to combine knowledge of vocabulary and of syntactic, semantic, and pragmatic constraints with knowledge of acoustics, phonetics, and phonology to form an integrated speech understanding system, using the knowledge from those higher level linguistic constraints to compensate for acoustic and phonological indeterminacies.

Nash-Webber, B., "Semantics and Speech Understanding," BBN Report No. 2896 (1974).

Abstract

In recent years, there has been a great increase in research into automatic speech understanding, the purpose of which is to get a computer to understand the spoken language. In most of this recent activity, it has been assumed that one needs to provide the computer with a knowledge of the language (its syntax and semantics) and the way it is used (pragmatics). It will then be able to make use of the constraints and expectation which this knowledge provides, to make sense of the inherently vague, sloppy and imprecise acoustic signal that is human speech.

Syntactic constraints and expectations are based on the patterns formed by a given set of linguistic objects, e.g. nouns, verbs, adjectives, etc. Pragmatic ones arise from notions of conversational structure and the types of linguistic behavior appropriate to a given situation. The bases for semantic constraints and expectations are an a priori sense of what can be meaningful and the ways in which meaningful concepts can be realized in actual language.

We will attempt to explore two major areas in this paper. First we will discuss which of those things that have been labeled "semantics", seem necessary to understanding speech. From the opposite point of view, we will then argue for speech as a good context in which to study understanding. To illustrate these points, we will begin by describing, albeit briefly, how semantics is being used in several recent speech understanding systems. We will then expand the generalities of the first section with a detailed discussion of some actual problems that have arisen in our attempt to understand speech.

Makhoul, J., "Linear Prediction: A Tutorial Review," IEEE Proceedings special issue on digital signal processing, (to appear April 1975).

Abstract

This paper gives an exposition of linear prediction in the analysis of discrete signals. The signal is modeled as a linear combination of its past values and present and past values of a hypothetical input to a system whose output is the given signal. In the frequency domain, this is equivalent to modeling the signal spectrum by a pole-zero spectrum. The major part of the paper is devoted to all-pole models. The model parameters are obtained by a least squares analysis in the time domain. Two methods result, depending on whether the signal is assumed to be stationary or nonstationary. The same results are then derived in the frequency domain. The resulting spectral matching formulation allows for the modeling of selected portions of a spectrum, for arbitrary spectral shaping in the frequency domain, and for the modeling of continuous as well as discrete spectra. This also leads to a discussion of the advantages and disadvantages of the least squares error criterion. A spectral interpretation is given to the normalized minimum prediction error. Applications of the normalized error are given, including the determination of an "optimal" number of poles.

The use of linear prediction in data compression is reviewed. For purposes of transmission, particular attention is given to the quantization and encoding of the reflection (or partial correlation) coefficients.

Finally, a brief introduction to pole-zero modeling is given.

Makhoul, J., "Linear Prediction in Automatic Speech Recognition," invited paper, IEEE Symposium on Speech Recognition, Carnegie-Mellon University, April 15-19, 1974, in _Speech Recognition: invited papers presented at the IEEE symposium_, D.R. Reddy (ed.), Academic Press (in press).

Abstract

This paper describes the recent applications of linear prediction to automatic speech recognition. Linear prediction is presented both as a spectral smoothing and a spectral modeling technique in which the signal spectrum is modeled by an all-pole spectrum. The method allows for the modeling of selected portions of a spectrum, for arbitrary spectral shaping in the frequency domain, and for the modeling of continuous as well as discrete spectra (such as filter bank spectra). Linear prediction is then compared to traditional analysis-by-synthesis techniques for spectral modeling.

Different parametric representations of the all-pole spectrum are introduced and compared for the purpose of speech recognition. These include the predictor coefficients, autocorrelation, spectrum, cepstrum, and reflection coefficients. The log area ratios are then proposed as a possibly optimal representation if a simple distance measure is used in the classification. A different approach to classification is also presented, where the distance measure is given in terms of a log likelihood ratio.

Recently developed parameters based on linear prediction for the purpose of feature extraction are given. These include formants, two-pole model parameters, spectral spread (a measure of the spectral dynamic range), and the first predictor and autocorrelation coefficients. An energy-independent spectral derivative is also proposed.

Nash-Webber, B., "Semantic Support for a Speech Understanding System," Proc. IEEE Symposium on Speech Recognition, CMU (April 1974).

Abstract

One function of the Semantics component of SPEECHLIS, the Bolt Beranek and Newman (BBN) Speech Understanding System, is to gather evidence for hypotheses it has made regarding the content of an utterance, as well as to evaluate the hypotheses made by other components. Another is to produce a representation of the utterance's meaning. Specifically, this involves forming consistent, meaningful collections of words which match regions of the speech waveform, and evaluating and interpreting the possible syntactic structures built of them. This paper

discusses the data structures and organization of SPEECHLIS semantics and how they are directed to the above two tasks.

Nash-Webber, B. and M. Bates, "Syntactic and Semantic Support for a Speech Understanding System," Presented at the 11th Annual Meeting of the Association for Computational Linguistics, Ann Arbor, Michigan, 1-2 August 1973.

Abstract

Six modular components knit together by a control strategy compose the BBN Speech Understanding System. These components are acoustic analysis, lexical retrieval, word matching, syntax, semantics and pragmatics. The syntactic and semantic components serve several roles. Their initial function is to select syntactically and semantically well-formed sequences of words from a lattice of possible word matches determined by the acoustical processing and lexical retrieval components of the system. They are also responsible for predicting words which may have been missed by the lexical retrieval routines but which are syntactically or semantically motivated by words that have already been found.

Under the direction of the control strategy, syntax and semantics are responsible for building and refining THEORIES. A THEORY is a hypothesis about a partially understood utterance -- the words it comprises and their syntactic and semantic organization. Many theories may be active at any time during the processing.

The syntactic component is structured around a parser capable of parsing either to the left or to the right, with provision for parsing in the face of discontinuous constituents. The data base of the semantic component is an associative net which is used both for answering requests and for noticing words in the lattice of word matches which are semantically relevant to a given THEORY. The semantic component also contains case information for verbs and nominals, which is used by syntax to test the semantic hypotheses expressed in a theory.

This paper will describe the structure of the syntactic and semantic components and aspects of their operation and interaction with each other and with the other components of the system.

Schwartz, R. and J. Makhoul, "Where the Phonemes are: Dealing with Ambiguity in Acoustic-Phonetic Recognition," Proc. IEEE Symposium on Speech Recognition, CMU (April 1974).

Abstract

Errors in acoustic-phonetic recognition occur not only because of the limited scope of the recognition algorithm, but also because certain ambiguities are inherent in analyzing the speech signal. Examples of such ambiguities in segmentation and labeling (feature extraction) are given. In order to allow for these phenomena and to deal effectively with acoustic recognition errors, we have devised a lattice representation of the segmentation which allows for multiple choices that can be sorted out by higher level processes. A description of the current acoustic-phonetic recognition program in the BBN Speech Understanding System is given, along with a specification of the parameters used in the recognition.

Rovner, P., J. Makhoul, J. Wolf and J. Colarusso, "Where the Words Are: Lexical Retrieval in a Speech Understanding System," Proc. IEEE Symposium on Speech Recognition, CMU (April 1974).

Abstract

Automatic speech understanding requires the development of programs which can formulate hypotheses about the content of an utterance and attempt to verify them. One example of such activity in the BBN Speech Understanding System (SPEECHLIS) is the use of information from a feature analysis of the sampled speech signal to propose and evaluate word matches which cover portions of the input utterance. Words proposed by higher level components are also verified against the feature analysis. It is at this interface between acoustic transcription and word matches that knowledge about the vocabulary, phonemic spellings, phoneme similarity, and phonological rules is represented and applied. The representation and use of such knowledge in the SPEECHLIS system is described.

Bates, M., "The Use of Syntax in a Speech Understanding System," Proc. IEEE Symposium Speech Recognition, CMU (April 1974).

Abstract

When a person hears an English sentence he uses many sources of information to assign structure and meaning to the utterance. One of these sources, syntax, is concerned with the goal of producing a consistent, meaningful, grammatical structure for

the sentence. The exact type of structure produced is not as crucial as the process of building that structure, because the speech environment has inherent problems, which make the parsing of speech a much more complex task than the parsing of text. For example, lexical ambiguity, caused by variations in articulation and imperfect or imprecise phoneme recognition, would lead to a combinatorial explosion in conventional parsers. This paper describes the design of the BBN speech parser with emphasis on the reasons for using the formalism of Transition Network Grammars and on the interaction of the syntactic component with other parts of the system. A detailed example is given to illustrate the operation of the parser.

Makhoul, J., "Spectral Analysis of Speech by Linear Prediction," IEEE Trans. on Audio and Electroacoustics, AU-21, No. 3, pp. 140-148 (June 1973).

Abstract

The Autocorrelation method of linear prediction is formulated in the time-autocorrelation and spectral domains. The analysis is shown to be that of approximating the short-time signal power spectrum by an all-pole spectrum. The method is compared with other methods of spectral analysis such as analysis-by-synthesis and cepstral smoothing. It is shown that this method can be regarded as another method of analysis-by-synthesis where a number of poles is specified, with the advantages of non-iterative computation and an error measure which leads to a better spectral envelope fit for an all-pole spectrum. Compared to spectral analysis by cepstral smoothing in conjunction with the chirp z-transform, this method is expected to give a better spectral envelope fit (for an all-pole spectrum) and to be less sensitive to the effects of high pitch on the spectrum.

The normalized minimum error is defined and its possible usefulness as a voicing detector is discussed.

APPENDIX C

WHERE THE PHONEMES ARE:
DEALING WITH AMBIGUITY IN ACOUSTIC-PHONETIC RECOGNITION*

Richard Schwartz

John Makhoul

Abstract

Errors in acoustic/phonetic recognition occur not only because of the limited scope of the recognition algorithm, but also because certain ambiguities are inherent in analyzing the speech signal. Examples of such ambiguities in segmentation and labeling (feature extraction) are given. In order to allow for these phenomena and to deal effectively with acoustic recognition errors, we have devised a lattice representation of the segmentation which allows for multiple choices that can be sorted out by higher level processes. A description of the current acoustic/phonetic recognition program in the BBN Speech Understanding System is given, along with a specification of the

----

*This paper was presented at the IEEE Symposium on Speech Recognition at Carnegie-Mellon University April 15-19, 1974.

parameters used in the recognition.

I. INTRODUCTION

One approach to automatic speech recognition begins the recognition process by attempting to divide the utterance into segments which are hypothesized to be single phonemes. The identity of each segment is then partially or completely determined by feature extraction or LABELING. Since segmentation and labeling are interdependent, the above process must be iterated to obtain reasonably accurate recognition. In this approach, segmentation errors such as missing and extra segments will arise not only because of the limited nature of an automatic algorithm, but also because of the inherent ambiguity of the acoustic signal. In general, it is not possible to identify segment boundaries with absolute certainty, nor is one sure of the exact phoneme that the segment represents [6,15,23]. Klatt and Stevens [21] have illustrated the types of acoustic variation that a single word can undergo depending on the context. Such variations can lead to segmentation and labeling errors if the only source of knowledge available is the acoustic signal. In this paper we shall illustrate the types of ambiguities that exist in analyzing a speech signal, and then outline the method we have adopted to deal with this problem in the BBN Speech Understanding System (SPEECHLIS) [48]. In addition, we give a brief escription of our current acoustic/phonetic recognition program (APR).

II. AMBIGUITIES IN THE SPEECH SIGNAL

Below are a few examples that illustrate the types of ambiguities that are found in the speech signal.

(a) A short dip in energy can be interpreted in several ways. For example, fricatives often have a short dip in energy at the start and end of frication. Also, a short nasal is often marked by a short drop in energy. Therefore, a dip in energy between a vowel-like sound and a fricative could be just a segment boundary, or a short nasal as in the word "answer".

(b) A silent segment followed by a noisy segment can be either a plosive followed by a fricative, or the whole sequence can be an aspirated plosive.

(c) Certain formant transitions can be interpreted as merely transitional, or as distinct phonetic segments. Broad [6] gives an example where the schwa in the word "away" in "we were away" looks just like a typical formant transition.

(d) Unstressed tense vowels often tend to look like their stressed but lax counterparts. Thus, the formants of the [i] in "pretty good" can look like a stressed [I].

Signal ambiguities, such as the examples given above, can lead to segmentation and labeling errors. Such errors occur also as a result of normal but unpredictable local variations in the signal, which frequently degrade the performance of recognition programs. There are, of course, also the usual errors due to insufficient knowledge. All these errors combine to make recognition based on acoustics alone very difficult.

Segmentation errors appear in the form of missing or extra segments. Labeling errors cause the wrong phoneme to be identified with a particular segment. Both types of errors can make it difficult for the correct word to match [37]. In our system, a small change in the quality of the APR makes a large change in the performance of the entire system. If an APR is required to come to a single decision at every point (i.e. produce a linear string of single phoneme segments), then segmentation and labeling errors could often be fatal. Such errors might be tolerated by the rest of the system if there is a small vocabulary and/or a limited syntax, from which to draw constraints. But if these constraints are not stringent enough, and a single segmentation is desired, then the APR must perform extraordinarily well to yield good overall recognition. It is clear that in general such accuracy in acoustic recognition is unlikely. One must be able to generate alternate choices so that the probability of correct recognition is increased. This is discussed below.

III. VAGUENESS IN RECOGNITION

The solution that we have adopted to deal with ambiguities in the signal and with segmentation and labeling errors is to introduce a certain amount of vagueness into the recognition process.

Vagueness in labeling is accomplished by allowing more than one phoneme to represent a segment. This increases the chances of having the correct phoneme appear in a segment label. However, this also means that the number of possible word matches [37] in each part of an utterance will also increase.

Vagueness in segmentation is implemented by allowing more than a single segmentation of any region of the given utterance. Instead of having only a sequence of adjacent segments, we now have the possibility of overlapping segments. The resulting segmentation forms what we call a SEGMENT LATTICE (to be described under <u>Segmentation and Labeling</u>; see also [37]). Again, this vagueness in segmentation increases the likelihood of finding the right words. However, many other words are found in addition.

It is desirable to have the correct words which are provided by the solutions described above, but the problems of dealing with a large number of extra words can be a very heavy burden on the system. Not only will there be an increase in computation but the problem of evaluating the different combinations of words can become very difficult. Therefore, one must be able to adjust vagueness thresholds to keep a workable balance between vagueness and correctness of segmentation and labeling.

One solution is to include with each segment, and with each phoneme in a segment label, a confidence measure of that being the correct path (sequence of segments) or phoneme. Most APR's use some sort of scoring algorithm to choose a path or a label. If the scores correlate well enough with reality to be used as a basis for a decision, they are also valuable as a mechanism for dynamically varying the number of choices during lexical retrieval [37]. In other words, by setting thresholds to be used with the scores, this system can simulate vagueness in a variable way. The question of how many paths through an utterance to allow is an efficiency matter. One would clearly not want to keep around information about all the possible paths. However, as long as the scores assigned to the paths are meaningful, keeping more paths around does not increase vagueness. It merely makes the system more flexible.

IV. ACOUSTIC PHONETIC RECOGNITION IN SPEECHLIS

The APR component in the current BBN Speech Understanding System consists of two basic sections: parameter extraction, and segmentation and labeling. The parameter extraction component operates on the speech signal at regular intervals and produces a set of parameters. These parameters are then used by the segmentation and labeling component to perform the actual feature extraction or recognition. The segmenter locates possible phoneme boundaries and constructs a lattice of optional segmentation paths. Each boundary has associated with it a confidence that it corresponds to an actual boundary. The labeler then describes each segment in the lattice in terms of acoustic features or phoneme classes, which are reduced to a small set of possible phonemes. Also associated with each segment is a measure of confidence that the correct description was found.

A. Parameter Extraction

The analog speech signal is sampled at 20 kHz into 12 bit samples and then normalized to 9 bits. All further processing is done on the sampled data. Preemphasis by simple differencing is employed only to obtain an energy measure (ROD) and a derivative of the preemphasized spectrum (SDE).

Parameters are computed at the rate of 100 frames per second. For each frame, an FFT is computed on 20 msec of the signal (Hamming windowed). The spectral region from 5-10 kHz is used only once to obtain a measure of the energy in that region (ROH). All other parameters are obtained by applying a 14 pole SELECTIVE LINEAR PREDICTION [25] to the 0-5 kHz region of the spectrum. The following table describes the basic set of parameters used. (For details on parameters related to linear predictive analysis, see references [25,28,29]. Wolf (1973)].)

NAME	DEFINITION OR DESCRIPTION
R0	Energy in the 0-5 kHz region
R1	Normalized 1st autocorrelation coefficient. Also equal to the average of a cosine weighted spectrum.
R0D	Energy of the differenced signal = 2*R0(1-R1)
V	Normalized LP (linear prediction) error. Also equal to the ratio of the geometric mean of the LP spectrum to its arithmetic mean.
VP	-10 log V
TPF	Frequency of the complex pole-pair, using linear prediction with 2 instead of 14 poles[29].
R0H	Energy in the 5-10 kHz region
SD	Average absolute value of the change in the LP spectrum between two consecutive frames (in dB)
SDE	Average absolute value of the change in the pre-emphasized LP spectrum (in linear units)
P0	Fundamental frequency

Figure 1. Basic Parameters

There is a set of corresponding parameters which reflect the change in the values of the parameters over a single frame (10 msec). These parameters have the same name prefixed by a "D". Another set of parameters reflect the change in the parameters over 30-50 msec. These parameters have the suffix "S" (for "slow"). For example, along with the parameter R0 we also have the "difference" parameters DR0 and DR0S. In addition, the formants are determined from the poles of the LP model.

B. Segmentation and Labeling

The present segmentation and labeling component can be broken into several major phases. These phases are logically separate but sequential (ordered). In the present implementation, however, they are executed in parallel, with appropriate lags separating them so that the analysis of one phase can effectively use any results of the previous phases.

1. Segmentation

A piecewise linear approximation to the formants is used to indicate possible "formant boundaries". In the first phase of segmentation, for each frame the absolute value of each difference parameter is compared with a threshold related to the specific parameter. If the threshold is exceeded, a score corresponding to this parameter is added to a total score for the likelihood that there is a boundary at that frame. Parameters considered in this phase are: DVP, DRO, SD, DVPS, DROD, SDE, FMBDR, DROS, and DRODS, in decreasing order of importance.

The values of the thresholds are such that most frames will end up with a score of zero. However, when there is a boundary, there is usually more than one frame with a non-zero score. In the second phase of segmentation, adjacent non-zero frames within 40 msec are "merged" into one boundary, if there is no evidence of a short nasal stop at that point.

In the third phase of segmentation, a piecewise linear fit to the parameter ROD is used to find new boundaries. If one of these new boundaries is close to a merged boundary, then the time of the boundary is changed to that of the new one. If there is no nearby boundary, then a new boundary is created.

Since the above procedures tend to find many extra boundaries, those with lower scores are considered optional. At this point, a LATTICE of segments is formed to express the optionality.

The lattice structure makes it possible to express different paths (sequences of segments) describing the period between two points in the utterance. In the lattice structure shown below, the horizontal axis represents time, and the vertical lines represent segment boundaries. The numbers are used to identify unique segments. There are 3 ways to describe the period from A to B: (1-2; 3-4-2; 5-6-7), two ways to describe period B - C: (8; 10-11), and two ways to describe period C - D: (9; 12-13-14). In all, there are 3x2x2=12 ways to describe the period from A to D.

```
|--5--|-------6-------|---7---|
|                             |
|--3--|---4---|               |--10--|--11--|-12-|-13-|-14-|
|             |               |             |              |
|-----1-------|-------2-------|------8------|------9-------|
A                             B             C              D
```

Figure 2.
Example Segment Lattice

2. Labeling

The labeling procedure for each segment consists of comparing average values of parameters (over the central half of the segment) to thresholds for several features (see table below). The averages of adjacent segments and the change in each parameter over the segment are also considered. The table below shows how a high or increasing value of each parameter correlates with the different features. Opposing features are separated by slashes, so that the presence of the first implies the absence of the second. For example, a high total energy (R0) indicates a sonorant and a nonobstruent at the same time.

PARAM	DESCRIPTION	FEATURES AFFECTED
R0	Total Energy	Sonorant/Obstruent, Vowel/Nasal, Voiced/Unvoiced, Fricative/Plosive
R0D	Energy of Differenced Signal	(Same kind of evidence as R0)
R0H	Energy between 5-10 kHz	Obstruent/Sonorant, Fricative/Plosive, Vowel/Nasal
VP	Normalized Error	Sonorant, Nasal, Voiced
TPF	Frequency of 2-pole LP model	Fricative, Vowel/Nasal, Reflects tongue height of vowels between 200-800 Hz
R1	1st Autocorrelation Coefficient	Indicates lack of high frequency energy, not a Fricative
F0	Fundamental Frequency	Its presence indicates voicing
F1 F2 F3	First Three Formants	Give information about the place of articulation of vowels and glides.

Figure 3.
Labeling Parameters

Associated with each segment description is a segment confidence, which is a score that reflects the confidence that the correct phoneme is included in the label. It is related to the scores of its constituent features, which depend on the deviation of each of the pieces of evidence (mostly parameter averages) from their neutral points. If one of the feature decisions is close to its neutral point, no decision can be made reliably, so both options are kept.

An attempt is made to fit cubic polynomials to the formants of segments with high energy. Target formants determined from these cubics are compared against model targets for the 15 vowels and glides in our system. Included is a frequency normalization based on the fundamental frequency. The matching procedure takes into account the individual values of the formants as well as the values of the formants relative to each other. The resulting match scores are used (along with duration for glides and diphthongs) to select up to four phonemes for the segment label.

For those segments labeled as strident fricatives, the place of articulation is determined by a threshold on the two-pole frequency (TPF) computed at a point two thirds of the way into the segment.

3. ROD Dip Detector

After the basic segmenting and labeling is finished, a dip detector is applied to the parameter ROD to find additional boundaries. If these boundaries do not correspond to the existing boundaries, additional (optional) branches are added to the lattice, and the new segments are labeled in the normal manner. The times of these new boundaries were found to correspond very well with the hand labeled boundaries. Therefore, these new boundaries will, in the future, be used to adjust the time of the other boundaries.

4. Special Cases

There are some checks made which take into account certain phonological phenomena. Certain segment boundaries found toward the end of the sentence are ignored because of the tendency to stretch out the end of a sentence. A path in the lattice described as unvoiced plosive followed by unvoiced weak frication is bridged by an optional single segment labeled as unvoiced plosive. Long plosives are optionally split into two plosives. Two adjacent segments with identical labels are bridged with one segment. These and other similar rules take into account some of the inherent ambiguity in the acoustic waveform.

V. FUTURE SYSTEM

At this time statistical studies of the correlations between certain parameters and features are being carried out. The scores on segment boundaries or on phonemes within a label will be determined by probabilities based on these studies. In keeping with the philosophy held here, each segment label will consist of a score for each phoneme (36 in our present system). Then, depending on the application, the lexical retriever would use all phonemes with a score above a certain threshold to achieve the desired vagueness.

Acknowledgement

The intial version of our accoustic/phonetic recognition program was written by D. O´Shaughnessy of M.I.T.

APPENDIX D

Travel Budget Management Sentences

List all trips to California this year.

How many trips has Craig taken?

What is the round trip fare to Pittsburgh?

Is two hundred dollars enough for a four day trip to New York?

What is the registration fee?

When did Bill last go to Washington?

Change the number of California trips to eight.

Cancel the trip to Tbilisi.

What is the new total of budgeted trips?

What is the auto mileage rate now?

Can I split the charges on that trip between the <X> account and the <Y> account?

How many trips to California are budgeted for this fiscal year?

How much money remains in the travel budget?

How much would it cost to send three people to London for a week in July?

How many people are scheduled to attend the IJCAI conference?

If I send 3 people to Sweden, will there be enough money left to send 5 people to Pittsburgh?

Is John scheduled to go to Carnegie?

What is the projected amount in the travel budget for fiscal 75?

How many trips has Bonnie been on this year?

What is their total cost?

If we send five people to California for a week, how many can we send to the IJCAI?

How much does it cost to send someone to California for a week?

What trips did John take last year and how much did (each, they) cost?

How many trips to Washington are proposed for next year?

Will the amount of money left in our travel budget cover the trips which have been proposed?

How much is the deficit?

What is the surplus?

How many (week long, three day) trips to California can we afford?

I want to know what trips Bill will take this winter.

How much would it cost to spend two days in L.A. and one day at Univac?

What is the round trip air fare to Miami?

Am I going anywhere in late November?

Who will be away the week of April tenth?

Which conference is the most expensive?

Which conference will cost the most for all the people going?

Do we have enough money left for a trip to St. Louis for 3 days for 2 people?

How much would a trip to California for 4 days cost?

Where is the next ASA Meeting?

When is the next ASA meeting?

How much have we spent on trips to N.Y.?

How many west coast trips have we taken?

How much would it cost to send 3 people to London for one week?

What is the cost of a 3 day trip to Pisa?

How many people did we send to the ACL conference?

What was the average cost?

What's this charge of $350 to 11510?

Are we sending anyone to the ICCL meeting in Ottawa?

There is going to be a meeting of the Steering Committee in December at SDC.

We should plan to send 2 people to the next phonological rules workshop which will be sometime in November.

What is the total estimated charge to 11510 for all of the planned trips that are outstanding?

What is the actual charge of all the trips we have taken?

What is the cost of all the speech trips?

Suppose I send three people to Santa Barbara.

Then what would the total estimated cost be?

What trips do we have budgeted for the rest of this contract year?

OK forget the three people for Santa Barbara and make it just two again.

How much of the 11510 travel funds are already spent?

How much is committed?

Are you aware of the next ASA meeting in St. Louis?

Who are the participants from BBN that plan to attend?

What are the dates of the meeting?

What is Jerry Wolf's trip number for this meeting?

What job number is being charged for each participant?

Tell me everything about trip number 1936.

What trips have been taken since February?

How much did they cost?

Were they all budgeted?

Were there any trips budgeted for, which were not taken?

Show me the rest of this year's travel plans.

How much do we have left in the budget?

Does that include John Makhoul´s trip to Salt Lake?

Assume John´s trip cost $600.

Change the number of Pittsburgh trips to 8 and add Craig to the list of peopl‹ going.

Are we over-budgeted?

Did we under-budget for that trip?

Did we budget correctly for trip 3778?

Have we allowed for Bill´s trip to Crete in October in the budget?

What percent of the money we asked for did we actually get?

When was the last time we checked through the travel record?

What´s the state of this year´s travel budget right now?

Do you nave any information on John´s trip to Salt Lake City this past April?

Do you know about any trips after 1 July?

How much is there left in the budget now?

Who´s roing to IFIP?

The final cost of that trip was $56.66.

Cancel Rich´s trip to Monterey for June.

John Plans to be in France in July from the 20th to the 22nd.

What´s the cost of a trip to L.A.?

What trips did we have budgeted for the speech project as of September, 1973?

Which of those trips have already been taken?

How much total money did we get from Bert for speech trips?

How much did we ask for?

How much have we already spent?

What unanticipated trips have we taken that were not in the budget?

Give me a list of the remaining trips with their estimated costs.

What's the total of those amounts?

Where is the spring acoustical society meeting?

Suppose we send only 4 people to New York and 4 to the ACL meeting.

Hold on to that supposition.

Give me a breakdown of the expenses to send one person to London.

What would be the total budgeted amount for 4 people to New York, 4 to ACL, 2 to London, 1 to Stockholm, plus the other untaken budgeted trips to other places.

Give me the breakdown of the costs for a trip to Amherst.

Change the travel estimate to $10 for the bus.

Change the registration, etc. to $50.

What is the total budget figure now for the assumptions mentioned above?

How much did we spend during the first quarter on trips that were not budgeted?

Make a note that we will expect to spend three times that much on unanticipated trips during the next three quarters.

Change number of remaining trips to Pittsburgh to be 9.

Add trip to Pajarro Dunes, California for 2 people 4 days.

The estimated cost per person is: air fare $350, hotel, food, etc. $140, and car rental, $75.

Add 3 people to Santa Barbara for 3 days and estimate cost.

Now what is the estimated budget for the remaining three quarters under supposition 2?

How much money do we have left unspent?

What was the air fare between Boston and Los Angeles?

How many people did we send to Amherst?

Add a $30 surcharge for visa costs to the IJCAI.

How much time was there between the London and Stockholm conferences?

Give me my total travel costs for the year to date.

Can we afford an additional person to the ASA meeting in St. Louis?

What is the total amount we have budgeted for international meetings this year?

How many person days are left in our budget for west coast trips?

How does our current budget differ from our original?

Compare the estimated and the actual costs for each of the trips to the west.

Isn't John going to some conference in California?

Why is Bill going to California?

Who paid for my trip to IJCAI?

How many people are budgeted to go to Russia?