AD-783 062

SPEECH UNDERSTANDING RESEARCH: COLLECTED PAPERS, 1973-1974

Madeleine Bates, et al

Bolt Beranek and Newman, Incorporated

Prepared for:

Advanced Research Projects Agency

July 1974

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						
SPEECHLIS						
Speech Recognition						
Speech Understanding						
Speech Understanding Research						
Speech Understanding Systems						
Syntax						
Transition Network Grammars						

AD 783 062

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
none

2b. GROUP

3. REPORT TITLE
Speech Understanding Research
Collected Papers
1973-74

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

5. AUTHOR(S) (First name, middle initial, last name)
Madeleine Bates
John J. Colarusso
John I. Makhoul
Bonnie L. Nash-Webber
Paul D. Rovner
Richard M. Schwartz
Jared J. Wolf
William A. Woods

6. REPORT DATE
July 1974

7a. TOTAL NO. OF PAGES
~~38~~ 45

7b. NO. OF REFS
25

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. 2856
A.I. Report No. 17

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

13. 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.

45

BBN Report No. 2856
A.I. Report No. 17

Bolt Beranek and Newman Inc.

SPEECH UNDERSTANDING RESEARCH

Collected Papers

1973-74

W.A. Woods, principal investigator
Madeleine Bates
John J. Colarusso
John I. Makhoul
Bonnie L. Nash-Webber
Paul D. Rovner
Richard M. Schwartz
Jared J. Wolf

JULY, 1973

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

PREFACE

This volume is a collection of papers presented at the IEEE Symposium on Speech Recognition at Carnegie-Mellon University April 15-19, 1974. Taken together they represent a snapshot of the BBN speech understanding project as of approximately December, 1973. The project is still far from complete and in particular is not yet to the point where definitive conclusions as to the success of the techniques described can be reported. However, I believe that the present collection of papers serves a useful purpose in documenting the state of the project and the approach that we are taking.

W.A. Woods

TABLE OF CONTENTS

MOTIVATION AND OVERVIEW OF BBN SPEECHLIS: AN EXPERIMENTAL PROTOTYPE FOR SPEECH UNDERSTANDING RESEARCH

W.A. Woods
Bolt Beranek and Newman Inc.
50 Moulton St., Cambridge, Ma. 02138

Introduction

This paper describes a computer system under development at Bolt Beranek and Newman 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 [10] and by Klatt and Stevens' spectrogram reading experiments [2]. 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 [12] we have described a 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.

Acknowledgements

The work being described in this paper has been done by the members of the speech understanding project at BBN to whom I am deeply indebted. These are:

John Makhoul (Signal processing and acoustic feature extraction)

Richard Schwartz (Acoustic feature extraction)

Jerry Wolf (Phonological rules, signal processing software and hardware)

Tony Scandora (Hardware)

John Colarusso (Phonology)

Dennis Klatt (Advanced word verification techniques)

Craig Cook (Advanced word verification techniques)

Paul Rovner (Lexical retrieval and control)

Bonnie Nash-Webber (Semantics and pragmatics)

Madeleine Bates (Syntax)

Laura Gould (Vocabulary)

Early work on acoustic feature extraction was done by Doug O'Shaughnessy.

The credit for constructing the system which holds the promise and has achieved the performance described herein goes to them. Any errors or inadequacies in the current exposition are of course my own.

Whereas this paper gives an overview of the system and its motivations, separate papers in this volume give more detailed descriptions of the operations of individual components. [1,6,7,8,9]

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 [11,13], 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 have 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.

Knowledge gathering

In order to gain a concrete understanding of the types of interaction required in using higher level linguistic knowledge to augment the front end (acoustic) analysis of the speech signal, we used "incremental simulations" to begin experimenting with the speech understanding system before completing its construction by "implementing" its components as combinations of computer program and human simulators. From these simulations, the following general conclusions were reached:

1) 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.

2) 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.

3) The process of infering an interpretation from a speech signal is inherently non-deterministic in the sense that 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.

4) No a priori order of scanning the utterance (such as left-to-right) for word matches and syntactic and semantic processing will be adequate in general since any given word may be garbled in its pronunciation or phonetic analysis and we may depend on 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).

5) 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 the set of possibilities which could be tried to get an interpretation when one has not found one yet is open-ended. Examples 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 there toward less and less likely interpretations. This requires the use of decision criteria to evaluate the goodness of a word match, and weigh the alternatives of 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.

6) 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.

Components of the System

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 6 conceptually distinguishable components

1) 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.

2) a lexical retrieval program 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.

3) 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.

4) 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.

5) 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.

6) 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 inferring 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.

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 current system. The scope of the current paper, however, will necessarily require these descriptions to be brief. For more detailed descriptions of the individual components the reader is again referred to the individual papers in this volume [1,6,7,8,9].

Acoustic-Phonetic and Phonological Analysis

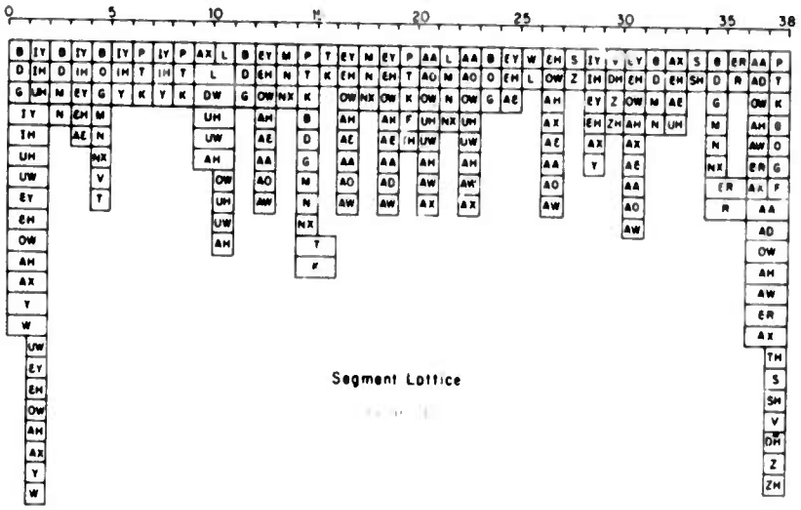
In the acoustic end of our system, the speech signal is sampled at 20 kHz and stored on a disc file. All subsequent analysis is performed on the digitized signal. Using our recently developed method of "selective linear prediction" [3,4] we perform a linear predictive (LP) analysis on the 0-5 kHz region 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 [5] 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 feature or 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 segments 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 current system.

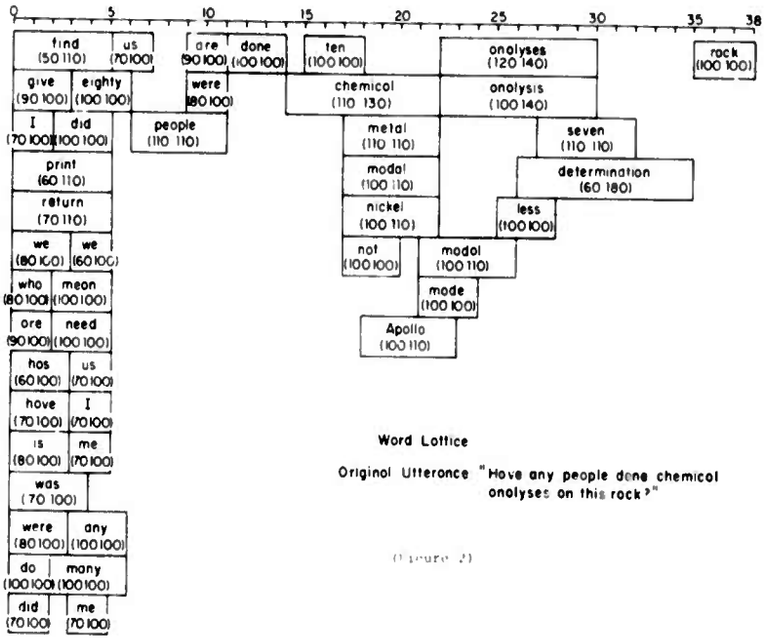
Higher Level Linguistic Constraints

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. 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



Segment Lattice



Word Lattice

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

(Figure 2)

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.

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:

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?

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.

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.

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 plosive 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 performed when the system is more fully developed to determine the most effective place to deal with this and similar problems.

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 our 250 word vocabulary, much less a 1000 word vocabulary.

The list of such questions which are being raised could go on and on, and would not be worth enumerating. However, the above list should be suggestive of the types of results which we hope to obtain.

A Sample of Current Performance

Issues of Evaluation

We have outlined the methodology and the current state of a project to develop advanced speech understanding capabilities by a process of continual incremental improvement of a system with initially crude capabilities in each of its individual 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 DWD-29.

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.

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 answers to these questions, we have run test cases which can serve as benchmarks. We will illustrate

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

with a brief summary of the syntactic and semantic processing of a sentence DWD24 ("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", "occuring", "with", "content", "contents", and many others. In the formation of one-word theories, 4 different matches of "contain" were combined into a single fuzzy word match, 4 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 contiquous 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, 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 although 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, 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.

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 behaviour 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.

Conclusions

We have presented a brief overview of the various components of the BBN speech understanding system 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 separate papers [1,6,7,8,9]. The components of the current system are but crude approximations of the components which we plan to evolve, 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 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 in mechanical speech understanding systems. 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 behaviour 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.

References

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

[2] 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, Ma. (1971).

[3] Makhoul, J., "Selective Linear Predictive Spectral Matching," presented at the 86th meeting of the Acoustical Society of America, Los Angeles, Ca. (also distributed as Speech Understanding Research Note 111, NIC 19951) (1973).

[4] Makhoul, J. and J. Wolf, "Linear Prediction and the Spectral Analysis of Speech," BBN Report No. 2304, Bolt Beranek and Newman Inc., Cambridge, Ma. (1972).

[5] 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. (1973).

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

[7] Rovner, P. et al., "Where the Words Are: Lexical Retrieval in a Speech Understanding System," Proc. IEEE Symp. Speech Recognition, CMU (April 1974).

[8] Rovner, P. et al., "Control Concepts in a Speech Understanding System," BBN Report No. 2703, Bolt Beranek and Newman Inc., Cambridge, Ma. (also Proc. IEEE Symp. Speech Recognition, CMU) (1974).

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

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

[11] 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).

[12] Woods, W.A. and J. Makhoul, "Mechanical Inference Problems in Continuous Speech Understanding," BBN Report No. 2565, Bolt Beranek and Newman Inc., Cambridge, Ma. (August 1973).

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

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

Richard Schwartz
John Makhoul
Bolt Beranek and Newman Inc.
50 Moulton St., Cambridge, Ma. 02138

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.

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 [1,2,4]. Klatt and Stevens [3] 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) [9]. In addition, we give a brief description of our current acoustic-phonetic recognition program (APR).

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 [1] 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 [8]. 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.

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 [8] 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 Segmentation and Labeling; see also [6]). 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 [8]. 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.

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.

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 (P0D) 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 (R0H). All other parameters are obtained by applying a 14 pole SELECTIVE LINEAR PREDICTION [5] 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 [5,6,7].)

NAME	DEFINITION OR DESCRIPTION
R0	Energy in the 0-5 kHz region
R1	Normelized 1st eutocorrelstion coefficient. Also equel to the average of e cosine weighted spectrum
R0D	Energy of the differenced signel = 2•R0(1-R1)
V	Normslized LP (linear prediction) error. Also equsl to the ratio of the geometric mesn of the LP spectrum to its arithmetic meen
VP	-10 log V
TPF	Frequency of the complex pole-pair, using linear prediction with 2 instead of 14 poles [ ]
R0H	Energy in the 5-10 kHz region
SD	Averege ebsolute value of the change in the LP spectrum between two consecutive frames (in dB)
SDE	Averege ebsolute velue of the chenge in the preemphesized LP spectrum (in linear units)
F0	Fundsmentel frequency

Figure 1: Bssic 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.

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.

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, DRØ, SD, DVPS, DRØD, SDE, FMBDR, DRØS, and DRØDS, 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 4Ø 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 RØD 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; 1Ø-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.

Figure 2: Exemple Segment Lattice

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 (RØ) indicates a sonorant and a nonobstruent at the same time.

PARAM	DESCRIPTION	FEATURES AFFECTED
RØ	Totel Energy	Sonorent/Obstruent, Vowel/Nasal, Volced/Unvolced, Frlcatlve/Plosive
RØD	Energy of Differenced Signal	(Same kind of evidence as RØ)
RØH	Energy Between 5-1Ø kHz	Obstruent/Sonorent, Frlcative/Plosive, Vowel/Nasal
VP	Normalized Error	Sonorent, Nasel, Volced
TPF	Frequency of 2-pole LP model	Frlcatlve, Vowel/Nasal Reflecte tongue height of vowels between 2ØØ-8ØØ Hz.
R1	1st Autocorrelation Coefficient	Indicates lack of high frequency energy, not e Frlcative
FØ	Fundamental Frequency	Its presence indicates volcing
F1 F2 F3	First Three Formants	Give lnformetion about the place of erticuletion of vowele end glidee.

Figure 3: Labelling 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.

RØD Dip Detector. After the basic segmenting and labelling is finished, a dip detector is applied to the parameter RØD 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.

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.

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 initial version of our acoustic-phonetic recognition program was written by D. O'Shaughnessy of M.I.T.

References

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

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

[3] 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., June 1971.

[4] 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.

[5] Makhoul, J., "Selective Linear Predictive Spectral Matching," presented at the 86th meeting of the Acoustical Society of America, Los Angeles, Calif., Oct. 30 - Nov. 2, 1973. (Also, distributed as Speech Understanding Research Note 111, NIC 19951).

[6] 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., Aug. 1972.

[7] 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, Mass., Sept. 1973.

[8] Rovner, et al., "Where the Words Are: Lexical Retrieval in a Speech Understanding System," Proc. IEEE Symp. Speech Recognition, CMU, April 1974.

[9] Woods, et al., "Motivation and Overview of BBN SPEECHLIS: An Experimental Prototype for Speech Understanding Research," Proc. IEEE Symp. Speech Recognition, CMU, April 1974.

WHERE THE WORDS ARE: LEXICAL RETRIEVAL IN A SPEECH UNDERSTANDING SYSTEM

Paul Rovner
John Makhoul
Jared Wolf
John Colarusso
Bolt Beranek and Newman Inc.
50 Moulton St., Cambridge, MA. 02138

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.

Introduction

A central problem in automatic speech understanding is how to select words for consideration as possible components of an utterance. If there are too many words to consider in each region of the utterance, then not only will the processing requirements tend to explode, but also the evaluation procedures can become untractable. Therefore, in order to treat the problem of speech understanding effectively, one must develop experience and insight into how to perform word selection while restricting possible combinatoric explosions.

The information available for word selection includes the vocahulary and how its words are pronounced, the syntactic, semantic, and pragmatic constraints of the task domain, the acoustic transcription (which includes segmentation and labeling), and knowledge about the ways in which the pronunciation of words can vary (phonological rules). 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. For certain such systems, possible words and partial word sequences can be enumerated (in a "top-down" manner) before considering the acoustic transcription, and then ordered on the basis of how well they match the acoustic transcription. The BBN speech understanding project[5] has chosen to develop a system for tasks in which such constraints are not strong enough to so limit the sets of possible words in the early stages of the understanding process. Instead, in a "bottom-up" manner, information from the acoustic transcription is used in an initial phase of hypothesis formation to suggest words which match well. These words are then sent to syntax and semantics for consideration.

Word selection occurs in SPEECHLIS at the interface between acoustic-phonetic programs which construct the acoustic transcription[4] and syntactic, semantic, pragmatic and control programs which combine word matches into tentative hypotheses about the meaning of the utterance[1,2,3]. The programs that perform word selection have two tasks: to use the acoustic transcription to propose words which could have been spoken (Lexical Proposal), and to evaluate how well a proposed word matches the acoustic information (Lexical Matching). The term "lexical retrieval" will be used to represent these two tasks. This paper describes the way in which lexical retrieval fits into the SPEECHLIS system, the strategies for Lexical Proposal and Lexical Matching, and the representation and use of phonological rules.

Lexical Retrieval in SPEECHLIS

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".

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 phoneres 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 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 labelled 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.

The Vocabulary. The vocabulary is represented as a set of words (currently 250), each having a set of its most likely pronunciations as lists of phonemes and syllable boundary markers. On the average, there are about two pronunciations represented for each word in the vocabulary. Associated with each phoneme is an estimate of the probability that it will be absent in an actual pronunciation of the word. Each vowel has 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.

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 labelled 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.

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 is to take account of such knowledge as these programs 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 analytic (as opposed to generative) phonological rules. Each rule has three components:

a) a template describing the necessary context to be searched for in the segment lattice.

b) 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.

c) 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 the 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. A predicate is an arbitrary Boolean function of three arguments: a phonemic spelling, the phoneme position within the spelling at which the new branch is being considered, and a pointer to the new branch in the lattice. A pointer to the rule's predicate is attached to each new branch when the branch is added to the lattice. This pointer is used by the lexical matcher to access and apply the predicate. The predicate returns true if it accepts the use of the branch in the word match or false if it rejects it.

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 (implicitly) 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 whenever a standard phonemic spelling is being considered by the lexical matcher.

An example of a phonological rule is the Nasal Deletion Rule. In its generative form, it is: "A nasal consonant can be deleted if it occurs immediately after a vowel and immediately before a nonnasal consonant with the same place of articulation." This rule says, for example, that in the word "sample" the [m] may be deleted (and the preceding vowel will be nasalized). It is implemented analytically as: "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 existing vowel segment by a two-segment branch consisting of the vowel followed by a nasal. Attach a predicate (described below) to the nasal segment." (If such a branch already exists,

then no new branch need be added.) The predicate requires that the nasal not be word initial, and it checks that the preceding phoneme (of the phonemic spelling) is indeed a vowel (and not a non-vowel which matched via a similarity), and that the nasal and the following consonant match in place of articulation.

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 to be examined by syntax, semantics, and pragmatics are entered into a WORD LATTICE (such a lattice is illustrated in Figure 2). In this figure, for example, the word "mean", spelled [M IY N], matches from 2 to 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 that word on the basis of the length (number of phonemes) of its phonemic spelling.

Usage

The overall control strategy for SPEECHLIS, starting from an acoustic transcription which has been expanded by the analytic phonological rules, is first to perform a scan 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. An attempt to find acceptable word matches at the beginning of the utterance from a set of likely sentence-initial words then occurs. Any such word matches are added to the word lattice. The system then enters a phase of tentative 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 support. As the system then attempts to verify, enlarge, and combine these theories, the lexical retrieval programs are called upon to evaluate the matches of words which are proposed by syntax, semantics, and pragmatics. Examples of such proposals are: content words which are likely to be adjacent to a word being considered, function words which are likely to follow a sequence of words, 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 specification of constraints on the length of acceptable words, word match quality acceptance thresholds, and requirements that word matches begin or end at a specified boundary or in a specified region of the segment lattice. In addition, there are parameters for selecting among several strategies for searching and matching, including the consideration of word matches with missing or extra segments. These strategies are described below.

Strategies

Lexical Proposal

There are two ways in which words can be selected 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. The other method is the "unanchored" scan, in which a word spelling is proposed if it has a specified pair of adjacent phonemes anywhere in its spelling. A set of such phoneme pairs is computed for each pair of adjacent segments in the given region of the segment lattice. This set is the cross product of the phoneme sets representing the two segments. The unanchored method is currently being used in SPEECHLIS for the complete initial scan.

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; a 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. 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.

Performance and Future Work

Since the first version of SPEECHLIS has only recently been assembled, we are not yet able to present a thorough analysis of the lexical retrieval performance requirements for acceptable overall system performance. From the few 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.

2. The quality of overall system performance depends greatly on the quality of lexical retrieval performance. The payoff of improvements in lexical retrieval performance will be high.

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. In addition to a continuing effort to improve the programs that perform segmentation and labeling, a program is being developed in which speech synthesis techniques will be used to construct a general representation of the expected acoustic parameters for a given phonemic spelling. These will then be matched against the parameters which were extracted from the real speech signal, and a score which represents the quality of the match will be computed. Depending on how well this "word verification" program performs, it will be used either to augment or replace the current lexical matching programs.

To further develop our experience and insight into how to perform lexical retrieval, statistics gathering experiments are being designed to evaluate 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 detection of those phonemes which one would expect to be "robust" (e.g. stressed vowels and strident fricatives).

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.

The problems of dealing with larger vocabularies (over 1000 words) and more elaborate phonological knowledge are imminent. One of our goals is to develop an understanding of how our lexical retrieval techniques should change as the system grows.

References

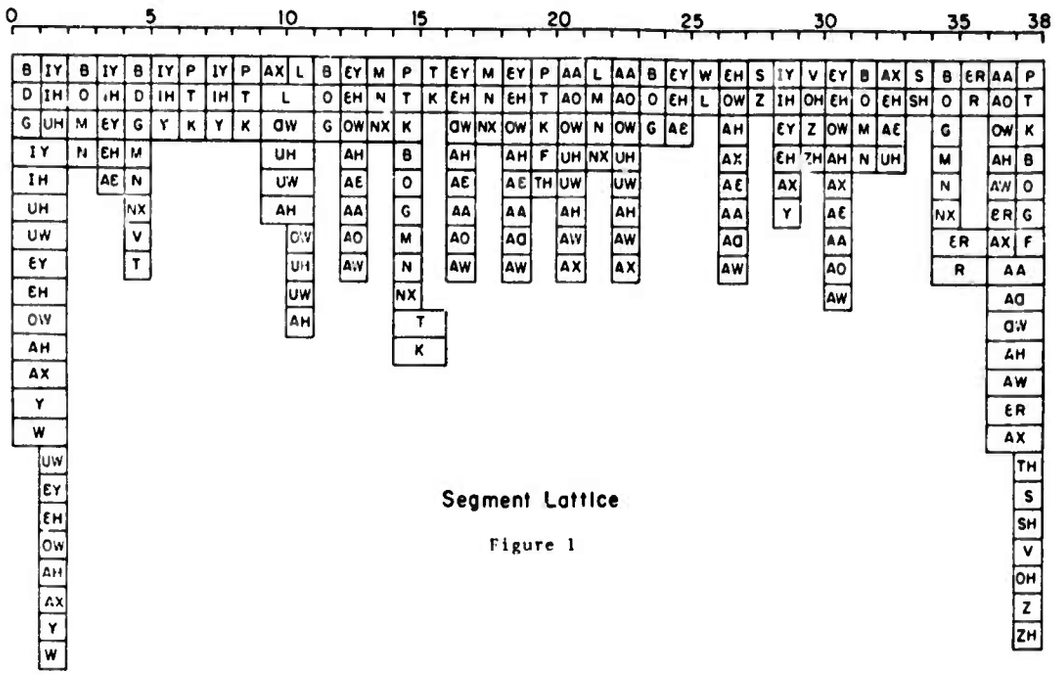
[1] Bates, L., "The Use of Syntax in a Speech Understanding System," Proc. IEEE Symp. Speech Recognition, CMU, April 1974.

[2] Nash-Webber, B., "Semantic Support for a Speech Understanding System," Proc. IEEE Symp. Speech Recognition, CMU, April 1974.

[3] Rovner, P. et al., "Control Concepts in a Speech Understanding System," BBN Report No. 2703, Bolt Beranek and Newman Inc., Cambridge, Ma. (also submitted to IFIP Congress 74).

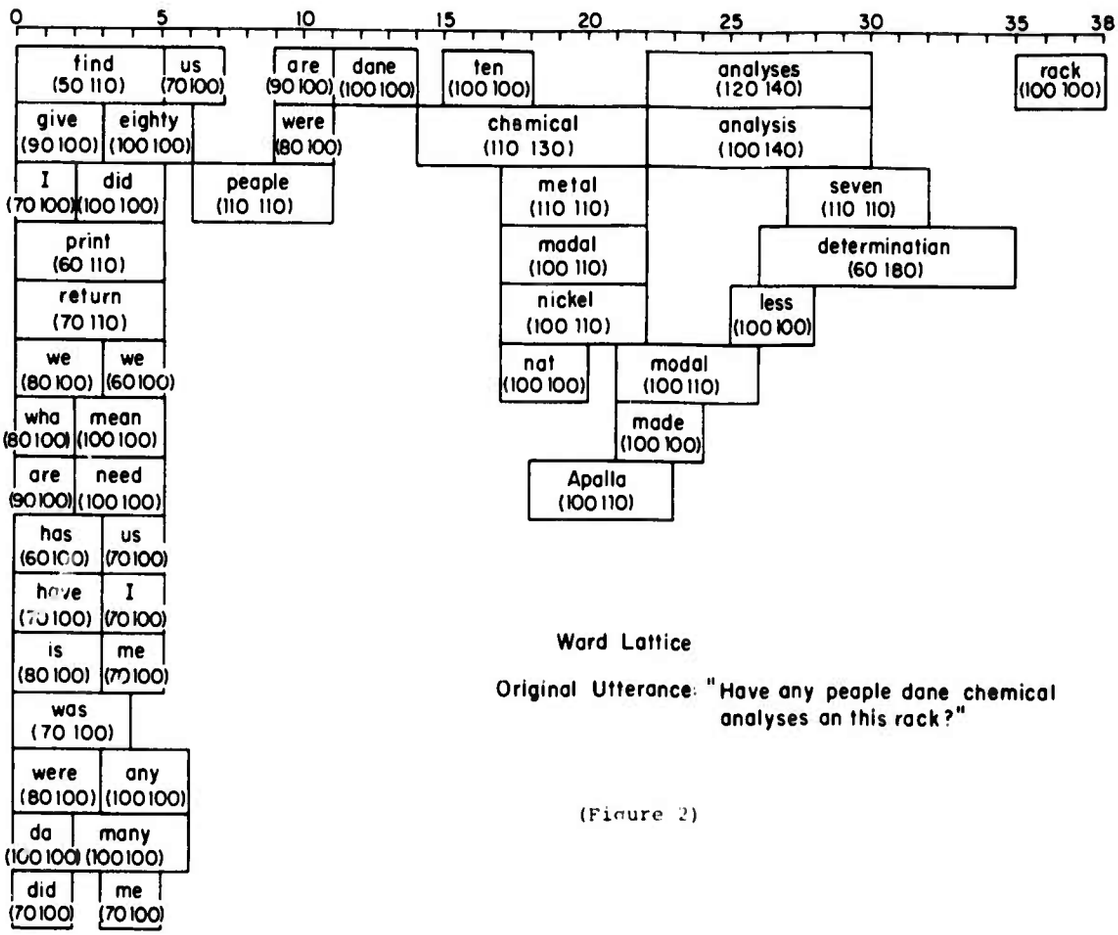
[4] Schwartz, R. and J. Makhoul, "Where the Phonemes Are: Dealing with Ambiguity in Acoustic-Phonetic Recognition," Proc. IEEE Symp. Speech Recognition, CMU, April 1974.

[5] Woods, W. et al., "Motivation and Overview of BBN SPEECHLIS: An Experimental Prototype for Speech Understanding Research," Proc. IEEE Symp. Speech Recognition, CMU, April 1974.



Segment Lattice

Figure 1



Ward Lattice

Original Utterance: "Have any peaple dane chemical analyses an this rack?"

(Figure 2)

CONTROL CONCEPTS IN A SPEECH UNDERSTANDING SYSTEM

Paul Rovner
Bonnie Nash-Webber
William Woods
Bolt Beranek and Newman Inc.
50 Moulton St., Cambridge, Ma. 02138

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).

Introduction

Speech understanding, whether done by people or by computers, demands great funds of knowledge. This is due to the inherent imprecision, variability and complexity of the acoustic signal into which all speech is encoded [1] . For example, the encoding of a word, spoken in running conversation, will be affected by its environment (the words surrounding it), its importance to the message (stress and intonation), its speaking rate, and its speaker. It is an apparent circular dilemma of acoustic-phonetics that one cannot precisely identify contextual effects without first identifying the context. However, we believe that by applying other sources of knowledge to an initially uncertain and imprecise acoustic transcription, in order to hypothesize possible higher contexts, this circularity can be broken.

In this paper, we are concerned with the problem of how to control the application of various sources of knowledge to this problem. A framework of concepts, data objects, queues, and programs is presented in which strategies for forming and evaluating hypotheses about the meaning of an utterance may be implemented and studied. One such strategy is described, and an example of its performance is given. A Speech Understanding System (SPEECHLIS) being developed at Bolt Beranek and Newman provides the environment for this work, and derives much of its structure from this control framework. Though it is not our purpose here to discuss in detail the design of SPEECHLIS, it will be useful to the reader to know that it contains several knowledge sources as components -- acoustic-phonetic, phonological, lexical, semantic, syntactic, and pragmatic.

A listener does not just passively accept speech: he actively uses all his knowledge to structure uncertain and incomplete cues from the acoustic signal into a grammatical, meaningful and appropriate utterance. Sources of knowledge which are available to a listener include:

1) The acoustic-phonetic properties of the language -- Knowledge of the correspondence between physically varying parameters of the speech signal and the basic phonetic elements of the language (phonemes).

2) The vocabulary -- One presumes that an English utterance will consist of a sequence of English words, interspersed, perhaps, with pauses and non-speech sounds. A vocabulary constrains the possible sequences of speech sounds and the set of words which might fit a particular sequence.

3) Phonological rules -- These rules specify allowable or characteristic variations in the pronunciation of words or phonemes in particular environments.

4) The syntactic structure of the language -- A sound sequence which is heard as the word sequence "in other samples" will not be heard as "in of a samples", since the latter is ungrammatical.

5) The set of concepts and relations that are meaningful to the listener -- A sound sequence which is understood as "close the doors" will not be understood as "close the daws", since birds don't close.

6) Pragmatic considerations (knowledge of the current context or situation) -- A similar sound sequence may be heard as "close the Dewers" in a room in which the only thing open is a bottle of scotch.

Much of the above knowledge is specific to a problem domain. For our automatic speech understanding effort at BBN, we have chosen the task area of an existing natural language question-answering system (LUNAR) for the Apollo 11 moon rocks [4], which answers questions such as:

> How many breccias contain more than 10% anorthosite?
>
> In which samples was titanium found?
>
> Give me all references to olivine twinning.

In doing so, we have been able to draw upon our knowledge of lunar geology and question-answering system characteristics developed during work on that system. The LUNAR system operates with a lexicon of around 3500 words. As of this writing, SPEECHLIS is operating with a 250 word

lexicon, with a larger one of about 1500 words in preparation.

Overview of the Control Framework

Data Objects

The control framework that we will discuss assumes the existence of programs which have access to various sources of knowledge. For example, acoustic-phonetic and phonological 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.

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 2 to 5 in the lattice, while the word "any", spelled [ni] 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.

In addition to waiting for new information (by setting monitors), the higher level components can actively seek it out. 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.

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:

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

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

3) 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.

4) 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 [3] 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 currently user independent, 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 discrepency 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.

A Control Strategy

Within the framework of word matches, theories, evaluation mechanisms, etc., a preliminary control strategy is undergoing computer implementation. 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. 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 inflexional 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.

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 [2]. The word lattice shown schematically in Figure 2 was constructed from it 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 this 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:

1) "give" (initial position) and "chemical analyses"
2) "give" (initial position) and "modal analyses"
3) "give" (initial position) and "chemical analysis"
4) "print" (initial position) and "chemical analyses"
5) "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:

1) "have ... done chemical analyses" and "people"
2) "have ... done chemical analyses" and "rock"
3) "give ... chemical analyses" and "me" (following "give")
4) "give .. chemical analyses" and "us" (following "give")
5) "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:

1) "have ... people done chemical analyses" and "rock"
2) "have ... done chemical analyses ... rock" and "people"
3) "give me ... chemical analyses" and "rock"
4) "give us ... chemical analyses" and "rock"
5) "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. (However, this ability to arrive at the same theory from several directions is necessary since it allows us to put together incomplete structures, irregardless 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 found not contribute anything to these good theories, and they were sent off to syntax.

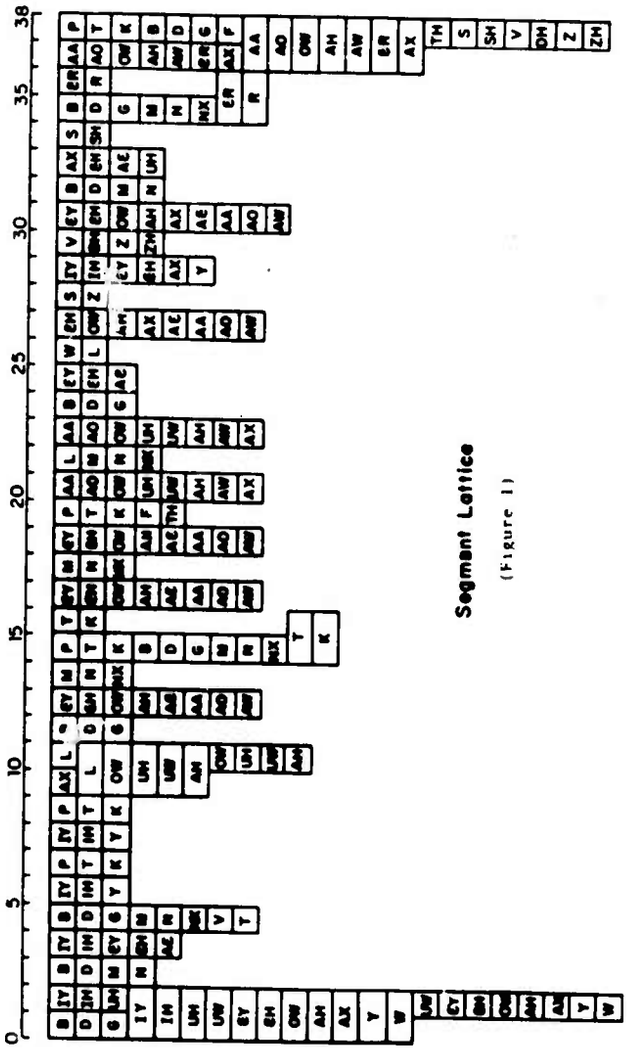
Syntax noticed the determiner "any" in the word lattice which could precede "people" syntactically, and it created an event which would refine the first theory with the word match for "any". In addition, syntax proposed determiners before "rock", since none occurred in the word lattice. This and additional 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, sufficient criteria for accepting the theory "Have any people done chemical analyses on this rock" as a correct understanding of the utterance.

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 going 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 use the technique of "incremental simulation" [5], in which a person simulates a part of the system which is not yet formulated to gain insight into how it might work.

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.

Other problems will arise from our imminent transition to a larger vocabulary and projected transition into different task domains. The attempt to solve all these problems will test the adequacy of our control framework in dealing with a world of uncertain and incomplete information.



Segment Lattice

(Figure 1)

References

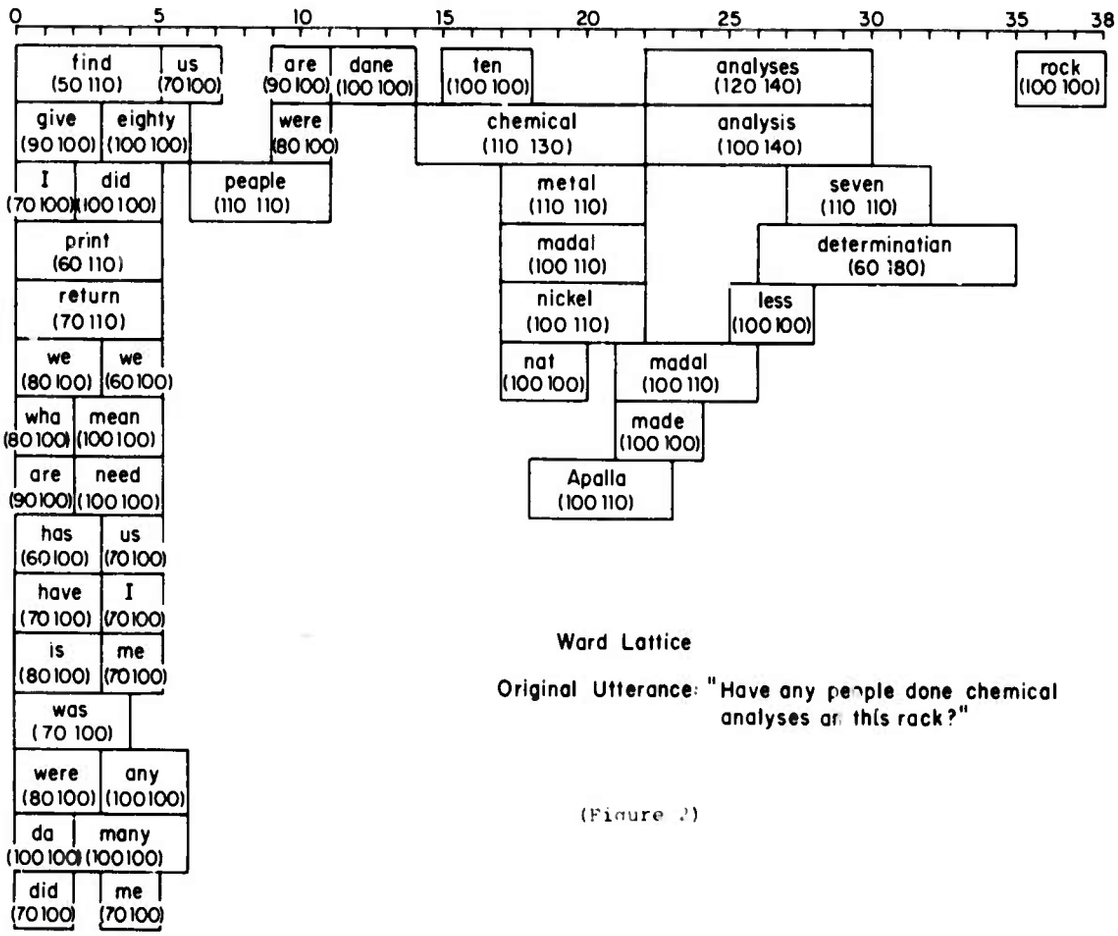
[1] Denes, P. and E. Pinson, The Speech Chain, Bell Telephone Laboratories, Murray Hill, New Jersey (1963).

[2] 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, Ma. (April 1972).

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

[4] Woods, W.A., R.M. Kaplan, and B. Nash-Webber, The Lunar Sciences Natural Language Information System: Final Report, BBN Report 2378, Bolt Beranek and Newman, Cambridge, Ma. (June 1972).

[5] 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).



Ward Lattice

Original Utterance: "Have any people done chemical analyses at this rack?"

(Figure 2)

THE USE OF SYNTAX IN A SPEECH UNDERSTANDING SYSTEM

Madeleine Bates
Bolt Beranek and Newman Inc.
50 Moulton St., Cambridge, Ma. 02138

Introduction

This paper will address four questions: (1) What makes the parsing of speech significantly different from the parsing of text? (2) What role does syntax play in speech understanding? (3) What strengths and weaknesses do existing methods of parsing text have in light of the answer to the previous questions? (4) How does the BBN speech parser cope with the problems presented?

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. Due to contextual effects, in order to uniquely identify a word in speech using acoustic parameters it may be necessary to know the words around it, but in order to identify those words, their context, including the original word, may be needed. The only way to break this cycle of impossibility is to allow considerable ambiguity in the word identification process. 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, as in the sequence "wait/weight four/for/fore the bare/bear," 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 of all this 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 1). 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.

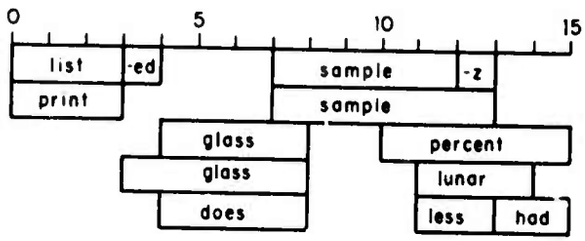


Figure 1. A partial word lattice

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.

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 2) 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.

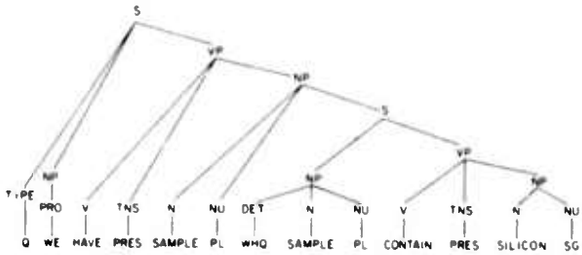


Figure 2. A deep structure for "Do we have samples which contain silicon?"

Text parsers are designed on the assumption that the words given as input will in fact 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 of predictions which can be made depends on the nature of the grammar being used and the amount of context which is taken into account when making the predictions.

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,p320; 2]), 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,p314] 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 the 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 (TNG) 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.

For a complete description of TNG's and a text parser using them, see [4] and [5]. 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 3 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.

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.

The BBN Speech Parser

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 Woods' paper in this volume [6]. 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 1. Only words of more than three 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 a hypothesis about the content of the utterance. For the remainder of this paper, 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(-z)]          [contain][silicon]
[print]     [ sample   ]
0     3     7     12    13        16       22       29
```

When the parser is given a theory to process, it processes the islands of words in it from left to right and attempts to create for each island of words 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 different register contents associated with the transition -- although not necessarily one for each possible context because sharing of register information reduces the number of sets required as we shall see below.

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

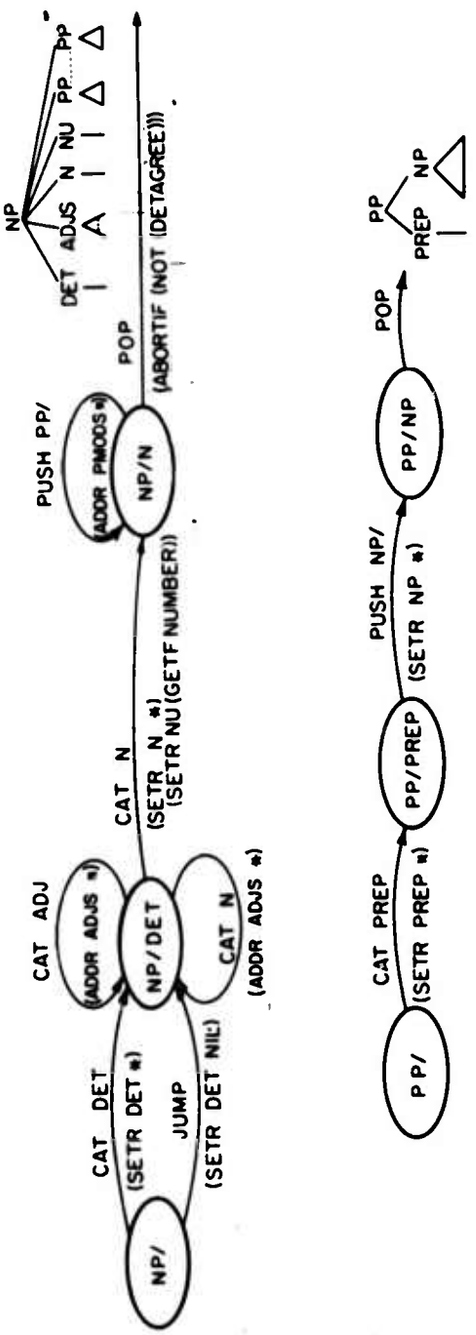


Figure 3. Small transition network grammar

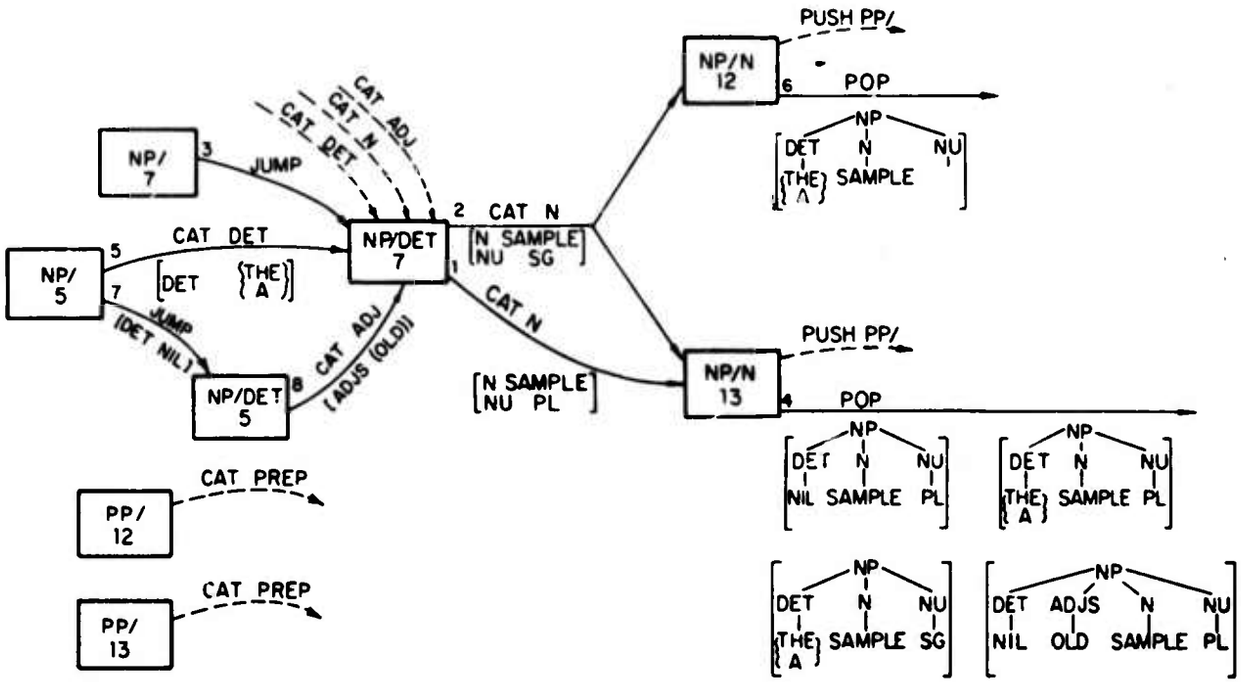


Figure 4. Map of transitions and configurations

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.

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 4 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. 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 3), 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.

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, and the registers N and NU are set by the actions on the arc. 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 4) in order to be able to process an event should one occur.

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.

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. Currently all open configurations are 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 number of registers set with unknown values, 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. 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 transition 4 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 are allowed to extend the path through that word.

The rest of Figure 4 shows the transitions and additions to the POP transitions which would be created for two events, one for the two determiners "the" and "a" and then one for the adjective "old". Notice that a register may contain a set of alternative contents, and that one or more of these alternatives may be selected for use by a later action. 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.

There are several features of the parser which the above example does not show. For example, if an action on an arc requires the contents of a register which is not set, a special symbol which means "unknown" is used in place of the desired value. POP transitions are prohibited from building structures containing an unknown value. When a transition is made which joins to the left end of a path which contains incomplete registers, new register sets are added to relevant transitions with copies of the unknown registers replaced by the correct values, and any pending POP transitions which have their register lists completed by this procedure may construct their results.

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 4 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.

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 register information when 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 wherever possible.

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 constricted 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.

Much more work remains to be done, particularly in the areas of semantic guidance and the inclusion of prosodic information, but we have established a framework which will allow for considerable experimentation with various strategies. We expect the system to serve as a tool to help us learn about the relationship between syntactic knowledge and the understanding of natural language.

References

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

[2] Earley,J., "An Efficient Context-Free Parsing Algorithm," CACM, 13:2 (Feb. 1970), 94-102.

[3] Nash-Webber, B., "Semantic Support for a Speech Understanding System," Proc. IEEE Symp. Speech Recognition, CMU, April 1974.

[4] Woods, W.A., "Transition Network Grammars for Natural Language Analysis," CACM 13:10 pp. 591-606 (Oct. 1970).

[5] 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).

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

SEMANTIC SUPPORT FOR A SPEECH UNDERSTANDING SYSTEM

Bonnie Nash-Webber
Bolt Beranek and Newman Inc.
50 Moulton St., Cambridge, Ma. 02138

Summary

One function of the Semantics component of SPEECHLIS, the 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.

Introduction

Psychologists have demonstrated that it is necessary for people to be able to draw upon syntactic and semantic information in their understanding of speech: the acoustic signal they hear is so imprecise and ambiguous that even a knowledge of the vocabulary is insufficient to insure correct understanding. For example, Pollack and Pickett's experiments [3] with fragments of speech excised from eight-word sentences and played to an audience showed that 90% intelligibility was not achieved until a fragment spanned six of the eight words, and its syntactic and semantic structure were becoming apparent. Similarly, the apparent impossibility of building a "phonetic typewriter" (a machine for taking dictation and producing English text) or of extending systems capable of single-word recognition to ones capable of recognizing continuous speech seems to imply that this ability to draw on syntactic and semantic information is necessary for computers too. Without making any claims about how a person actually understands speech, this paper will present some kinds of semantic knowledge and the ways in which they can help a listener to understand an utterance. An initial attempt to organize, represent and use such semantic knowledge in SPEECHLIS will also be described.

Kinds of Semantic Knowledge

Semantic knowledge of the names of familiar things and of models for forming the names of new ones permits a listener to expect and hear words which make sense by naming things which he knows. For example, knowing the words "iron" and "oxide", their meanings, and that a particular oxide (or set of them) may be specified by putting the name of a metal before the word "oxide" can enable a listener to hear the sequence "iron oxides", rather than "iron ox hides" or even "Ira knocks sides".

Knowledge of how concepts can be expressed linguistically enables the listener who expects to hear a particular concept to tune himself for words and phrases which can realize it. For example, all of the following are possible realizations of a concept the listener might be expecting:

samples with no sodium

samples which do not contain sodium

samples in which sodium does not occur

sodium-free samples

Knowledge of lexical semantics (models of how words are used) enables the listener to predict and verify the possible surface contexts of particular words. For example, "contain" names a two place relation. When it is used in an active sentence, its subject is to be understood as a location or holder, and its object, as something capable of being located or held. In a passive sentence, the active object becomes the passive subject and the active subject or location is realized in a prepositional phrase headed by "in".

Every lunar breccia contains silicon. (Active)

Silicon is contained in every lunar breccia. (Passive)

This knowledge also enables a listener to hear things which make sense rather than things which don't. For example, the following are two possible transcriptions of the same utterance. The first is more likely, since the subject of the second, though an acceptable noun phrase, cannot be understood as a location or holder.

Washington's tin contains traces of sulfur; Oregon's does not.

Washing tungsten contains traces of sulfur; Oregon's does not.

Semantic knowledge of the meanings conveyable by different syntactic structures enables the listener to hear cues to syntactic structure which might otherwise be lost. Syntactic structure is often signalled by very weak acoustic cues such as small function words like prepositions and determiners. The knowledge of what semantic relations can meaningfully hold between two concepts can often help in recovering the syntactic cues which signal them. For example, the preposition "of" can get lost in an utterance of "analyses of ferrous oxide". Yet the only meaningful relation between "analyses" and "ferrous oxide" that can be expressed, given this word order, demands that "ferrous oxide" be realized as a prepositional phrase headed by "of" or "for":

'analyses ferrous oxide" is meaningless. This enables a listener to hear the "of" which might otherwise have been lost.

The SPEECHLIS Environment

In the BBN Speech Understanding system (SPEECHLIS), an effort is underway to provide a framework in which the above-mentioned kinds of semantic knowledge may be represented and used to produce an appropriate semantic interpretation of an input utterance.

Before discussing the current embodiment of SPEECHLIS semantics in detail, it would be useful to describe briefly the environment in which it operates. [For a more detailed exposition of the SPEECHLIS world, see references 4 and 6.]

The three higher-level components comprising the system's knowledge of syntax, semantics, and pragmatics work to produce a syntactically sound, semantically meaningful and pragmatically felicitous reconstruction of the original utterance. Input to these components is a Word Lattice which is produced by acoustic-phonetic, phonological and lexical retrieval programs from an analysis of the input utterance. Entries in the word lattice are words which are found to be likely matches in regions of the speech waveform. Because there may be more than one such word match in any region, a lattice of alternative possible word matches results, rather than a single string. Associated with each such word match is a description of where it matched and how well. Initially, only words of three or more phonemes in length are included, since shorter words tend to produce possible matches everywhere.

A set of control programs determines the operational sequence of the other parts of the system - who does what when - keeping track of what has already been done and what is left to do.

The higher-level components work together to produce Theories. A theory is a hypothesis that a set of word matches belongs to an utterance. This set need not span the utterance, but may only cover parts of it. A theory contains information about how the word matches can fit together syntactically, semantically and pragmatically, as well as measures of how confident each component is in that theory. Various events may happen during the analysis of a theory which would tend to change the weight of the theory, to make SPEECHLIS more or less happy with it. For example, if no word could be matched just to the right of a given word match, we would be less certain about its being in the original utterance. On the other hand, were "sample" to match well to the right of a word match for "lunar", we would be more confident about both words being in the original utterance. Event Monitors are active agents set up by one of the higher-level components which watch for events, and create an appropriate Notice when one has occurred. Examples of semantic monitors and events will be found further on in this paper.

Organization and Use of Semantic Knowledge in SPEECHLIS

Organizational Factors

The sequence of words which lay behind the utterance for its speaker may not be its only reading for a listener: "his wheat germ and honey" could easily be heard as "his sweet German honey". In order for the listener to arrive at the same reading of the utterance as its speaker, he must be able to use whatever cues he can get from the speech signal to reconstruct the entire utterance. Moreover, the precision with which people speak varies, so that the strong cues - those which the listener feels he can most dependably trust - cannot be determined a priori.

Reconstructing the utterance starting from one of its parts requires models of possible utterance constituents and the ability to access these models starting from any part. These may be models of syntactically well-formed constituents - e.g. noun phrases - as well as models of semantically meaningful ones. A semantics system for speech understanding must not be constrained to accessing its semantic models in one way because that way may not be suggested by the available cues. This was not a strong factor in previous semantic systems designed for the automatic analysis of written text where the sequence of words in the input was known. For example, the semantic models in the BBN LUNAR system [5], built for NASA to answer written English questions about the Apollo 11 lunar samples, were templates on syntactic tree structures which specified selectional restrictions on their leaves and were classified and retrievable only by their head noun or verb. The programs which did the template matching could not begin to determine which templates might be applicable, and thus what other parts of the model to look for, without the identification of the head noun or verb. For speech applications, semantic information must be organized and accessible in ways that will enable the listener to make use of the strong cues he did hear, even if the head noun or verb is garbled or misheard.

Data Structures

The data structures of SPEECHLIS semantics have been designed to represent the kinds of semantic knowledge mentioned above in a way that allows flexible access. The two principal structures - a semantic network and case frame tokens - are discussed below.

SMALL SEMANTIC NETWORK

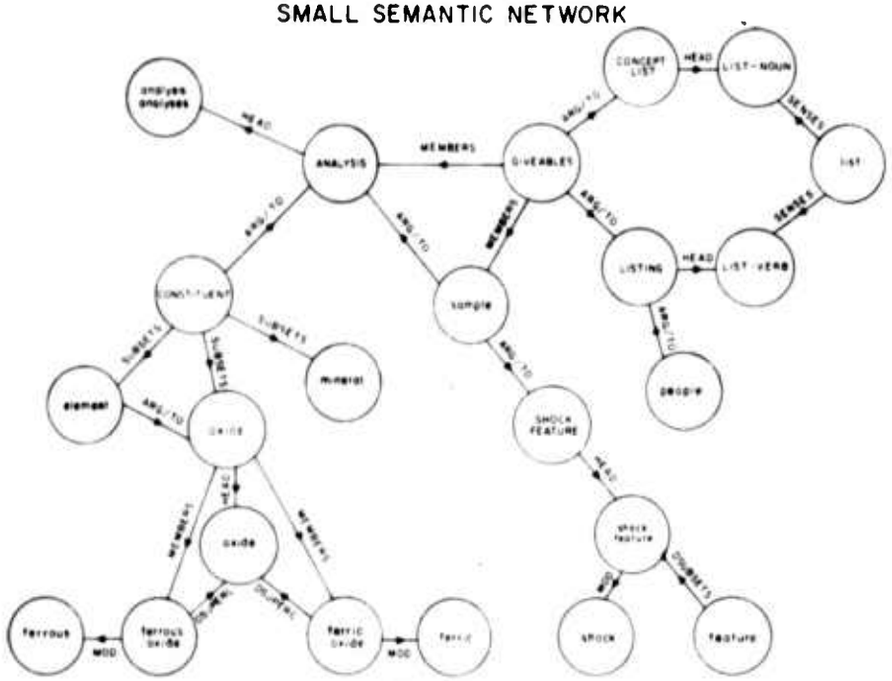


figure 1

The Semantic Network. A semantic network, a tiny piece of which is shown in Figure 1, represents the associations among words and concepts. (The names of nodes which do not correspond to specific English words are indicated in capital letters.) There are currently three types of nodes in the network. The first kind corresponds to concepts named by single English words like "ferric", "iron", and "contain".

The second type of node corresponds to concepts like "fayalitic olivine" which we refer to as "multi-word names". They are identified by the types of arcs entering and leaving them. Dsubset, dsubstuff, and dsuperc links say that one concept is a subset, substuff, or superconcept of another by definition. That is, while both fayalitic olivine and peridotite are types of olivine, only the former is so by definition. A mod link goes to the node which effects the subcategorization of the original concept. (The mod link does not specify how the meaning of the modifier affects the meaning of the modified concept. For example, the links between the nodes for "fayalitic olivine" and "fayalitic" and those for "principal investigator" and "principal" are both mod links, though the modifiers do not affect the nouns in the same way.)

Semantics uses its knowledge of multi-word names to propose additional words to the word matcher. That is, given a word match, the rest of a multi-word name of which its word is a part might have occurred in the original utterance, but be missing due to poor match quality. The effort the word matcher spends here depends on how necessary it is for the word match to be part of a multi-word name. For example, given a word match for "oxide", Semantics would ask the word matcher to look for "ferrous" or "ferric" to the left of "oxide", naming "ferrous oxide" or "ferric oxide". Given a match for "ferric" or "ferrous", Semantics would ask it to look much harder for "oxide", since neither "ferrous" nor "ferric" could appear in an utterance alone.

The third type of node represents a relation - a concept which takes arguments. For example, the relation named by "analysis" takes two arguments -- an instantiation of the concept CONSTITUENT, e.g. "iron", and one of the concept SAMPLE, e.g. "each breccia".

Semantics uses its knowledge of words, multi-word names, and relations to construct theories for meaningful sets of word matches. Given a word match, Semantics follows arcs through the network, looking for multi-word names and relations of which it or a concept that it instantiates may be a part. On each of the other components of the name or relation, Semantics sets monitors. Should an event occur in which a monitored component is instantiated and both general and specific conditions are met, the monitor creates a notice calling for the construction of a new, expanded theory.

To see this, consider again the network shown in figure 1 and a word match for "oxide". Semantics would find that "oxide" is one of the components of "ferrous oxide" and "ferric oxide", and would set monitors on the nodes corresponding to "ferrous" and "ferric" with the specific condition that any match for which a notice is to be created appear to the immediate left of "oxide". Word matches which trigger these monitors must also satisfy the general condition which disallows overlapping word matches.

Semantics would also find that oxides could be constituents of rocks and a constituent could be one argument to the relation named by "analysis" - "analyses", the other being the concept SAMPLE. (Note that a node which can be referred to by the root form of a word is also referred to by any inflected form.) Both nodes corresponding to "analysis" - "analyses" and to SAMPLE would be monitored. Subsequently, a non-overlapping word match for "analysis" or "analyses" or one which instantiates SAMPLE (e.g. "rock") would be seen by a monitor and result in the creation of a notice linking "oxide" with the new word match.

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 ANALYSIS than will a theory for "analysis" and "oxide". The event for the latter is given a higher weight than the former. (One is more certain that a particular relation has been expressed if one has heard its name mentioned rather than one or more of its possible arguments.)

Case Frame Tokens. The semantic network indicates the existence of relationships between concepts. Case frames [2] on the other hand describe how these relationships hold and how the relation may be expressed in an utterance. Associated with each relation is a case frame such as the one shown in Figure 2 for ANALYSIS.

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. (A case usually names an argument place in a relation, but we have extended its use somewhat to include the relation itself as a case, specifically the head case (NP-HEAD or S-HEAD). This allows a place for the latter's instantiation in an utterance, as well as the instantiations of each of the arguments.)

Among the types of information in the first part is a specification of whether a surface realization of the case frame will be parsed as a clause (REALIZES . CLAUSE) or as a noun phrase (REALIZES . NOUN-PHRASE). If as a clause, further information specifies which cases are possible active clause subjects (ACTIVSUBJ's) and which are possible passive clause subjects (OTHERSUBJ's). (While not usual, there are verbs like "break" which allow several possible cases to become the active subject, but the order in which they are chosen is determined a priori by which cases are actually present. Thus, the cases in ACTIVSUBJ are ordered, given the presence or absense of each case. However, there is no preferred order in selecting which case becomes passive subject, so the case names on OTHERSUBJ are not.) The first part of the case frame may also contain other information such as inter-case restrictions as would apply between instantiations of the object and goal cases of RATIO - 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 a Fillmore-type [2] case name or an abbreviation of one: "OBJ" for object, "AGT" for agent, "GOAL" for goal, etc.)

b) the way it can be filled - whether by a synonym for a concept (EQU) or by an instantiation of it (MEM), e.g. (EQU . SAMPLE) would permit "sample" or "lunar sample" to fill the case, but not "breccia" which refers to a subset of the samples.

c) a list of prepositions which could signal the case were it realized as a prepositional phrase (PP). If the case is not realizable as a PP, this entry will 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, will be derivable from context (ELLIP). For example, in "The bullet hit.", the object case - what was hit - will be derivable from context.

Tasks

Two tasks of SPEECHLIS Semantics have already been mentioned in the section on data structures: to propose additional words which might have occurred in the original utterance but were missing from the initial word lattice because of poor match quality, and to construct meaningful sets of word matches from a lattice of possible ones. A third task of Semantics is to evaluate the consistency of syntactic structures and semantic hypotheses.

Semantic Evaluation. As more word matches are included in a theory, Semantics represents its hypotheses about their semantic structure in case frame tokens. These are case frames which have been modified to show which word match or other case frame token fills each instantiated case.

The two case frame tokens in figure 3 represent 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 goal case (NP-GOAL) is filled by another case frame token representing "ferrous oxide". Another way of showing this is in the tree format of figure 4. There are a small number of syntactic structures that each possible set of cases can be realized as: here, the head case must correspond to the syntactic head and the goal case must be realized as either a prepositional phrase or 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. Notice that the only difference between the terminal strings of (a) and (c) is the presence of the preposition "of". Yet, this small word makes the difference between an acceptable syntactic structure and an unacceptable one.

As the syntactic component of SPEECHLIS (see [1]) builds structures, Semantics evaluates them against its hypotheses and assigns a score to them which depends on how many of its hypotheses are fulfilled and how much material on the syntactic tree violates or is not part of Semantics' hypotheses.

Other Tasks

Semantics has two other tasks in SPEECHLIS whose implementations are not far enough along to describe in detail. First, Semantics should guide Syntax to the most meaningful parse as directly as possible. That is, Syntax should not make random choices in places where Semantics has information that can be used to order the choices. This will be implemented via Syntax's ability to ask questions of Semantics on the arcs of the Transition Network Grammar [1], and will eliminate the need to wait until a well-formed substring with syntactic structure is created before getting a measure of meaningfulness from Semantics.

Finally, Semantics should transform the best theory (or theories) about an utterance into a formal procedure for operating on its data base in order to answer questions or to absorb new information. This is where "speech understanding" differs from "speech recognition". SPEECHLIS will not have to distinguish among best theories which mean the same thing (i.e. are mapped into the same formal procedure), though differing in the exact words they contain. Many of the interpretation methods that we used in the LUNAR system we expect to carry over into the speech world.

Current State of SPEECHLIS Semantics

Based on a vocabulary of approximately 175 content words on lunar geology and the names of the 43 Apollo 11 samples, a semantic network has been constructed, containing approximately 350 nodes. (The other 75 words in the SPEECHLIS vocabulary are function words - determiners, prepositions, auxiliaries, and conjunctions - whose meanings do not seem to be the types of things the current network can represent.) We have run the higher level components on only a small number of word lattices, so the following results are only preliminary impressions. In analyzing an utterance, each new theory seems to set, on the average, 5-6 monitors on nodes of the network. This is not so extraordinarily low, as a theory for a verb word match will only set monitors on the arguments to the relation it names, and the number of arguments to any relation rarely exceeds three. On the average, 4-5 event notices will be created during the processing of each theory. Many of these events are very unlikely, and experiments with pruning strategies - not creating unlikely events - seem to show good results, with, on the average, one notice being built per theory.

Many problems remain to be solved: for example, SPEECHLIS Semantics will be extended to larger vocabularies and larger semantic networks. It is not clear, for example, by how much the network would grow, were the vocabulary size to double, triple, or even quadruple, or were we to want to use the network for other tasks such as inference making. But we take for granted now the important role that Semantics must play in automatic speech understanding, so these and many other problems will have to be faced.

CASE FRAME FOR ANALYSIS

```
(( Realizes . Noun-Phrase) )
 ( Np-Head (Equ .14) Nil Obl)
 ( Np-Goal ( Mem . 1) (Of For) Ellip)
 (Np-Loc ( Mem . 7) (In For Of On) Ellip) )

Concept 14 - Concept of Analysis
Concept 1 - Concept of Component
Concept 7 - Concept of Sample
```

figure 2

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

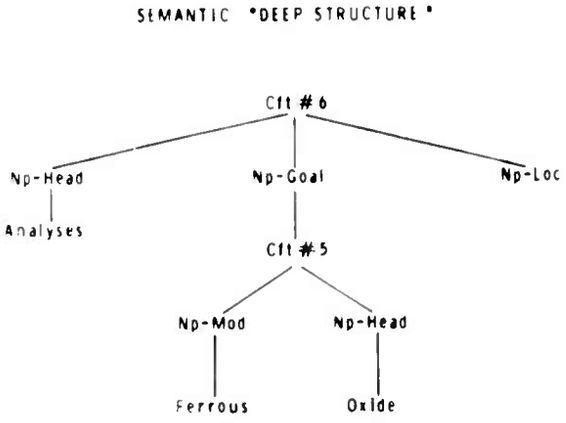


figure 4

References

[1] Bates, M., "The Use of Syntax in a Speech Understanding System," Proc. IEEE Symp. Speech Recognition, CMU, April 1974.

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

[3] Pollack, I. and J. Pickett, "The Intelligibility of Excerpts from Conversation", Language and Speech, VI, pp. 165-171 (1964).

[4] Rovner, P., B. Nash-Webber and W.A. Woods, "Control Concepts in a Speech Understanding System", BBN Report No. 2703, Bolt Beranek and Newman Inc., Cambridge, Ma. (1973).

[5] 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).

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

SYNTACTIC STRUCTURES

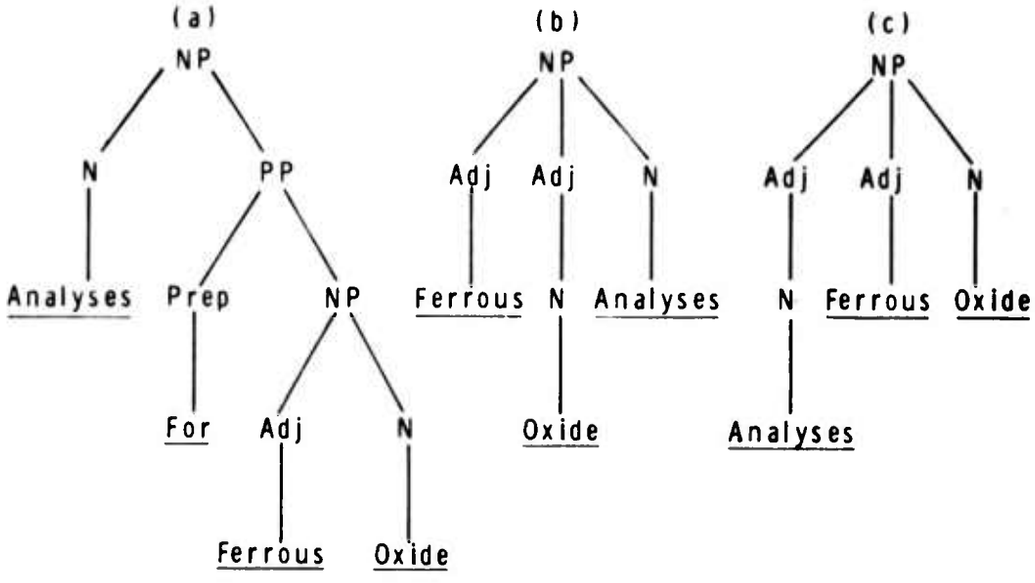


figure 5