Session 7 Natural Language:
Speech

SPEECH UNDERSTANDING THROUGH SYNTACTIC AND SEMANTIC ANALYSIS*

Donald E. Walker
Artificial Intelligence Center
Stanford Research Institute
Menlo Park, California

ABSTRACT

Stanford Research Institute is participating in a major program of research on the analysis of continuous speech by computer. The goal is the development of *a* speech understanding system capable of engaging a human operator in a natural conversation about a specific problem domain. The approach being taken is dlstinctive in the extent to which it depends on syntactic and semantic processing to guide the acoustic analysis. This paper provides a description of the first version of the system, emphasizing the kinds of information that need to be added for effective results.

INTRODUCTION

Stanford Research Institute is participating in a major program of research on the analysis of continuous speech by computer (see Newell et al., 1971) being sponsored by the Advanced Research Projects Agency (ARPA). The goal is the development of a speech understanding system capable of engaging a human operator in a natural conversation about a specific task domain. Our path toward this goal has been a characteristically "artificial intelligence" approach. We believe that many of the critical problems involved cannot be anticipated outside of the context of a functioning system. As a result, our first efforts were to build a preliminary version, using, where possible, available programs as components. Because of the critical role that we expect semantics to play in the final system, we chose, as a base, Winograd's programs for understanding natural language (Winograd, 1971). Accordingly, we accepted for our first task domain his simulation of the actions of a robot that knows about and can manipulate blocks of various shapes, sizes, and colors. The intent was to allow a person speaking to the computer to ask questions about the "blocks world," to give commands that would modify it, and to add information that would augment its structure.

During the first year of the project, we completed a first version of our system that did allow us to use syntactic, semantic, and acoustic data in the analysis of parts of spoken utterances. More importantly, we learned something about the problems of speech understanding. We already have begun work on a second version that will use a new parser, now under development, and that will involve a different task domain. This new system is still in the process of construction, although the parser is far enough along for us to present it in a companion paper (Paxton and Robinson, 1973). We have chosen to describe the first version in this paper, because we believe that the basic system concepts we are testing are illustrated there. Moreover, the problems we have identified are ones that we think are worth presenting to other members of the artificial intelligence community. Thoroughgoing solutions to these problems will require more than the resources available within our project at SRI or even those in the ARPA program as a whole. The problems involved In acoustic analysis are not examined here; only enough information is presented to clarify the nature of the approach being taken in the system design.

Most previous work on voice input lo a computer is referred to as "speech recognition" rather than as "speech understanding." (see Hill, 1971, and Lea, 1972). Research on speech recognition has aimed at providing an orthographic transcription of the sounds and words corresponding to the acoustic signal. The major emphasis in systems designed for that purpose has been on acoustic processing; some groups have developed pattern matching strategies, while others have tried to identify units phonetically or phonemically and to aggregate them info larger and larger units. While there have been some results with isolated words from relatively small vocabularies, extrapolation of these techniques to continuous speech has not been successful.

In contrast, research on speech understanding seeks to determine for spoken utterances the message intended in relation to the accomplishment of some task and in spite of indeterminacies and errors in the generation, transmission, and reception of an utterance. The processing of syntactic, semantic, and pragmatic information is considered essential, and a question-answering system may even be used as a major component.

There are a variety of approaches to speech understanding being taken by participants in the ARPA Program. It would be beyond the scope of this paper to sketch them out; however, descriptions are presented at this Conference of the work at Carnegie-Mellon University (Erman et al., 1973; Reddy et al., 1973) and at Bolt Beranek and Newman (Woods and Makhoul, 1973). Elsewhere, there are reports on the design of the System Development Corporation system (Barnett, 1972) and on the work at Lincoln Laboratory (Forgie, 1972a, 1972b). Some of these efforts concentrate on acoustic analysis of the speech signal, segmenting and labeling phoneme-like units that will be grouped into words—and more

The work reported herein was sponsored by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC04-72-C-0009 with the U.S. Army Research Office.

References are listed at the end of the paper.

complex grammatical structures—according to syntactic and perhaps semantic criteria. Others accept hypotheses from a number of sources, for example, acoustic, syntactic, and semantic, each of which may be checked against "the rest. Actually, the ARPA program is still in its early stages, and none of these systems—our own included—can be said to have established final design specifications, so specific contrasts arc hard to draw firmly. Moreover, different task domains would seem to respond differentially to one approach rather than another, a point that will be considered again later.

In the system we are developing at SHI, knowledge about the task domain, the grammar, and the current state of the analysis are used to constrain the selection of the. word or words that might be expected to be present at a particular place in the speech stream representing an utterance. The acoustic data for that location are analyzed to determine the degree of correspondence with each expected word by a program that characterizes its acoustic structure. When the presence of a word is confirmed, this information, in conjunction with the other sources of knowledge in the system, leads to the selection of another word for testing at the next place in the speech stream. Successive steps provide both a segmentation of the utterance into words and a specification of its syntactic and semantic structure. The distinctive aspects of the design are its strong dependence on syntax and semantics and its deliberate minimization of hypotheses generated solely on the basis of acoustic data.

The capabilities developed for the first version of the SRI speech understanding system were rudimentary, but it did predict words and tesi for their presence. More preprocessing of the acoustic data was done than we believe should be necessary. Acoustic characterizations were prepared for only a few words, so it has not been possible to step through a complete utterance. Other sources of knowledge are clearly desirable, for example, a model of the user and information about prosodies—stress, intonation, pauses. Nevertheless, the results of this first implementation were sufficient both to encourage us to continue this approach and to provide guidance for the revisions in progress.

AN INTEGRATED SYSTEM FOR SPEECH UNDERSTANDING

Introduct ion

In the first version of the SRI system for speech understanding, there were three major components: a set of procedures for syntactic and semantic analysis; programs for acoustic processing; and a word verifier routine that links the other two. Successive versions will include additional components and require major changes in all three of the present ones, as well as much more complex interrelationships. Nevertheless, this version of the system does illustrate an approach to speech understanding that is distinctive because of its dependence on syntactic and semantic processing.

The syntactic and semantic component, named Pintle, is a major modification (in ways described below) of Terry Winograd's system for procedural analysis of language (Winograd, 1971). A grammar—written as a set of programs—is combined with semantic routines that model changes in the arrangement of a set of blocks. A sentence constitutes a path through the grammar. Branching at choice points is determined by the order of the rules, by features on other constituents, and by semantic data. At the end of each branch in the parse tree is a set of words from a particular grammatical class (e.g., determiners, adjectives, nouns, verbs), from which a subset can be selected on semantic grounds.

The acoustic routines convert the recorded analog voice input to digital form. The digitized signal is then fed into a bank of digital filters, which make it possible to assign successive acoustic segments to one of a set of small, crude, but highly reliable classes. The signal also is processed by a more complex acoustic analysis procedure that identifies the frequency and amplitude for the ffrst three spectral peaks of the vowel-like sounds. The parametrized data from these two analyses arc stored in files,

The word verification routines take a set of words produced by Pintle and test each word against the acoustic data for a particular portion of the utterance. The result is a (possibly empty) subset of the words, ordered according to agreement with the acoustic data, with each word containing a pointer to identify its approximate endpolnt in the acoustic stream. Pintle takes the most likely word first and then proceeds on its path through the grammar to select the next set of words for processing by the word verifier. Testing this new set against the acoustic data begins at the point designated by the endpolnt for the word previously accepted. If none of the proposed words agrees with tha acoustic data, Pintle backs up to the most recent choice point and tries another alternative.

An example is considered next, and a more detailed description of each of the components is presented in Section III.

Understanding a Sample Sentence

A brief description of the "blocks world" problem domain used in the SRI system is necessary as background for the analysis of the sample sentence. Visualise a table containing a box and several objects of different sizes, shapes, and colors. There are five blocks (two red, one black, one green, one blue) and three pyramids (green, blue, and red); the box is white. The objects are arranged in a particular configuration in the computer representation of the scene, but the details of the arrangement are not necessary to understand the example. Commands given to a simulated robot arm cause it to move the blocks. Alternatively, the person Interacting with the system can ask questions or provide information that will augment or change the semantic structure of the world in some way.

The sentence to be processed is the following:

PUT THE BLACK BLOCK IN THE BOX.

It was recorded, digitized, and parametrized in advance, and the results stored in a file that was loaded into the system for the test. All of the steps involved in its analysis are presented as they occurred in an acutal demonstration. The capabilities shown reflect the state of the system as of December 1972. (The speech understanding system is implemented in BBN-LISP and is run on a PDP-10 computer under the Tenex Operating System. Lines prefixed by an arrow represent entries by the user.)

```
«# (PUT THE ($ 37) ($ *) IN THE ($ 134).)
```

Analysis of the sentence by Pintle is initiated. At the time this protocol was made, the word verifier did not have word functions available for the sets of words including PUT, THE or IN. Under these circumstances—and in general to allow more flexible testing of the system—it is possible to enter text to specify a word. For convenience, the word verifier checks the input text first to see whether any of the words in the set predicted has been typed in. Finding none, it will use the appropriate word functions, if they are available. If none are present, the words in the set will be rejected.

```
PUT
```

Pintle begins by looking for a major clause; branching along the imperative path, it looks for command verbs. The word PUT is among those in the set generated at this point, and it is found in the text input.

```
THE
```

Having found a verb, Pintle begins its search for a noun group by looking for a determiner, THE is confirmed from the text input,

```
BLACK
```

Having found a determiner, Pintle looks for an adjective. Since there is no text input, the predicted words are tested against the acoustic data. Beginning at a location 370 milliseconds into the utterance (the 37th 10-millisecond segment), the word verifier finds that BLACK corresponds to the acoustic data at the highest of four confidence levels; RED, GREEN, BLUE, and WHITE are rejected.

```
BLOCK
```

Pintle now looks for a noun to complete the noun group. It begins at the location in the acoustic data confirmed as the ending place for BLACK, the previous word accepted. This condition is specified by the asterisk in the initial entry for the sentence to be processed. The word verifier finds that BLOCK corresponds to the acoustic data at the highest confidence level; BALL, PYRAMID, and THING are rejected.

```
IN
```

With the noun group complete, Pintle looks for a prepositional phrase to complete the action denoted by PUT. IN is confirmed from the text input.

```
THE
```

Looking for a noun group to complete the prepositional phrase, pintle begins with the determiners. THE is confirmed from the text input.

```
BLACK
```

Pintle next looks for adjectives. Beginning at a location 1340 milliseconds into the utterance (the 134th segment), the word verifier finds that BLACK corresponds to the acoustic data at the third highest confidence level. RED, GREEN, BLUE, and WHITE arc rejected.

```
BOX
```

The location returned along with BLACK proves to be the end of the utterance. Consequently, BLACK is rejected because the string so ending would be ungrammatical; it also would be semantically unacceptable. Pintle backtracks and begins looking for nouns. BOX corresponds to the acoustic data at the highest confidence level; BALL, PYRAMID, and THING are rejected.

```
(PUT THE BLACK BLOCK IN THE BOX)
```

The sentence as identified. At this point the system returns a printout showing the grammatical structure of the sentence.

```
(CLAUSE MAJOR IMPER ACTV TRANSL)

     (VG IMPER)

          PUT (INF PAST VB TRANS VPRT MVB)

     (NG OBJ 0BJ1 NOLOC DET DEF NS)

          THE (DET NPL NS DEF)

          BLACK (ADJ)

          BLOCK (NOUN NS)

     (PREPG PLACE LOBJ)

          IN (PLACE PREP)

          (NG OBJ PREPOBJ DET DEF NS)

               THE (DET NPL NS DEF)

               BOX (NOUN NS)
```

Briefly summarized, the sentence is identified as an active imperative with the verb PUT, involving the movement of an object, THE BLACK BLOCK, to a location, IN THE BOX. (See Winograd, 1971, and Walker, 1973, for more extended discussions of the grammatical structure.)

This information is used to direct movements of the simulated robot arm in conjunction with the semantic definitions for the words and the determination of their relevance for the current state of the problem domain, as indicated in the following steps,

MOVETO 472 192 128

The arm moves to the black block, specified by the three coordinates given.

GRASP :B3

The arm grasps B3, the black block.

MOVETO 448 448 129

The arm with the black block moves to the box, specified by the three coordinates,

UNGRASP

The arm releases the black block inside the box.

OK

The system indicates that it has completed the action identified in its "understanding" of the input utterance.

Provisional Features of the Implementation

In anticipation of a more detailed discussion of the system components in the next section, it is probably reasonable to note here some characteristics of the analysis of the sample sentence that were temporary expediencies even in this version of the system.

The system was not totally on-line; i.e., it was not possible to speak directly into the system and to initiate processing accordingly. Analog-to-digital conversion of the speech signal cannot be performed on our PDP-10/15 computer facility yet, pending completion of the necessary software. Consequently, the signal was digitized on a PDP-11 and the resulting files transferred by tape to the PDP-10 for the rest of the acoustic analysis.

The FORTRAN files accessed by the word verification routines contained preprocessed data from both the digital filters and from the more complex analysis. Initially, we expected that in the final system we would only produce immediately on input the preliminary classification of acoustic segments provided by the digital filters. Spectral analyses and other complicated acoustic processing were to be performed only ae required to make the kinds of decisions necessary to distinguish among the predicted words in relation to the acoustic data. It now seems likely that complex analyses can be done in real time, so that we can get a richer parametric representation of the utterance. However, we still do not want to make specific classification decisions apart from the word verification procedure.

As noted in the analysis of the sample sentence, only a small number of word functions were written. Consequently, this version of the system was never used to process a complete sentence. The option of testing predicted words against textual, as well as acoustic, data proved useful for debugging the acoustic routines for particular sets of words. It also is useful in the absence of semantic and prosodic procedures for establishing constraints on paths through the grammar at the beginning of utterances, and, in particular, at the beginning of a dialog when no context has been established.

A final comment on the analysis embodied in the sample sentence is probably in order. We did not exercise this first version of the speech understanding system to any great extent. There were only a few word functions, and they were tested against only two speakers. The flow of control was primarily from the syntactic and semantic component to the acoustic. It was clear from the beginning, however, that useful information could pass in the opposite direction, not only from what a prosodic analysis might provide, but also from what might be expected to arise in the course of testing words against the acoustic data. In addition, we envisioned ways in which the word verifier, which in this version of the system processed one word at a time, could operate more efficiently on the whole set of predicted words in relation to the acoustic data, thus reducing the search space involved.

COMPONENTS OF THE SYSTEM

Pintle—Procedures for Syntactic and Semantic Analysis

Pintle, the syntactic and semantic component of this version of the speech understanding system, is based on the Winograd "Computer Program for Understanding Natural Language" (Winograd, 1971). It is a top-down system for linguistic analysis in which syntax, semantics, and inference are combined to direct the . processing of questions, statements, and commands. As implemented by SRI in BBN-L1SP, Pintle constitutes a substantial modification of Winograd's program.

In Winograd's work, as in most existing parsing systems, successive words from a typed input string guide the analysis. Since we proposed to use the parsing procedure to help segment and identify the words in the speech input, it was necessary to find other ways to control the generation of paths through the grammar. So syntactic and semantic constraints were established to influence successive choices, leading to the selection of a subset of the words of a particular word class. In what follows, the information available in the grammar for this purpose is presented, with an explicit identification of the assumptions required for effective use of the system in this context, and with some additional elaborations where appropriate.

Consider again the sample sentence discussed in the previous section, PUT THE BLACK BLOCK IN THE BOX.

Assuming that at the time this utterance is made in a hypothetical dialog of a user with the system it is reasonable to expect a command, the clause program would look for an imperative. Semantic information would be critical here, but knowledge about the user also would be helpful, and pronodic data—discussed further below—could provide useful guidance. Since imperative clauses generally start with verbs, the parser enters a verb group program looking for imperatives. Since imperatives are in infinitive form, only those verbs with that feature are identified. The result of this path through the grammar is a small set of imperative verbs, one of which may correspond to the first word of the utterance. We expect to be able to constrain the set of verbs further by additional semantic information—perhaps regarding what command might be appropriate at this point in the dialog. And information specific to a particular user should be possible to capture; for example, the frequent use of certain commands. However this verb group is constrained, the initial result is a set of words to check against the acoustic data.

Confirming one (or more) of the words from this inilial set might result in Pintle looking for *a* noun group, as is the case with the word PUT, which requires an object. The use of the word "might" is deliberate, to indicate the possibility of alternative choices. Identification of *a* different imperative. PICK, could result in Pintle looking first for the particle UP. Accepting PUT in the sample sentence, Pintle might begin the search for a noun group with a determiner. Since the sot of determiners is small, off of them could be predicted. However, they are difficult to distinguish acoustically, and it might be reasonable, on semantic grounds, to look only for a definite or only for an indefinite determiner, e.g., THE or A.

Finding a determiner, an adjective would be likely to follow. There are various classes of adjectives, and in English there is an ordering controlling the sequence in which they typically modify a noun. For instance, size adjectives precede color adjectives; e.g., DIG RED BLOCK but not RED BIG BLOCK. Again, in a dialog it would be reasonable at certain points to predict the amount of specificity required to identify an object on the basis of its qualities. Furthermore, some people may make things perfectly clear, while others are more sparing in their characterizations. Having models of the users could be valuable for providing this information. So, sets of adjectives would be checked against the acoustic data. Subsequently, and in a similar fashion, various paths among the nouns would be selected for testing. The kind of verb would influence the choice; verbs of manipulation call for nouns that represent manipulable objects. This information also could be used to influence the choice of an adjective in the prior search, limiting it to those adjectives appropriate to manipulable objects.

Continuing the parse beyond the noun group would lead to consideration of preposition groups because PUT requires a location. Identifying a place preposition would lead to a search for an object noun group, with decisions being made similar to those discussed for the preceding noun group. However, only those nouns that can have objects PUT IN them need to be considered.. In this manner, a set of predictions can be made regarding the sequence of sets of words likely to occur in the utterance.

The foregoing description presumes the accuracy of the initial predictions. In the sample sentence, however, the adjective initially found in the second noun group proved to be in error. Thus, backtracking and tracing down an alternate path were required to find the noun. An interpreter for PROGRAMMAR was added to provide a backtracking mechanism not available in Winograd's system (and not necessary for Winograd, see below). The interpreter made it possible to specify a set of alternatives at a particular point in the grammar and to try these in succession, backtracking automatically if the initial choice was not subsequently confirmed. This same mechanism made recovery possible following acceptance of a word that proved to be in error, as in the sample sentence.

The requirement for speech input (the absence of words with identifiable features in the input string) and the availability of the backtracking facility resulted in other modifications to Winograd's analysis procedure. Winograd tested to eliminate the least likely alternatives first, checking the longest possible constituent and cutting back when that failed. PROGRAMMAR, in his original version, returned the first successful analysis, having provided both syntactic and semantic guidance to make that a likely interpretation within the model of the "blocks world." Selective backup was possible in a particular situation, but it involved specifying a location to return to for alternative processing. With voice input, it is necessary both to test for most likely alternatives first and to have a more general mechanism for following alternative paths through the grammar in case of failure. What is needed further for speech understanding is the flexibility in the grammar to allow dynamic reordering of rules, depending on the state of the analysis at the moment. It would be desirable to be able to identify at any particular choice point in the grammar the alternatives that are possible. In Winograd's system, alternative choices could only be identified serially after failure of the predecessor.

Many more changes in the overall parsing strategy are needed to improve its ability to use syntactic, semantic, and—hopefully—pragmatic constraints to make accurate predictions about the words likely to be present at any particular place. In Winograd and in our first version, checking against the actual configuration of objects on the "blocks world" could be done only after a group had been parsed. Thus, in the sample sentence, both BALL and PYRAMID were tested against the acoustic data. However, there were no balls in that configuration (although the word was in the lexicon), and there were no black pyramids. Information of this kind can and should be used to influence the selection of words In a set as soon as it is relevant.

The introduction of new structures for managing semantic and pragmatic information could be expected to have major consequences for parsing. It certainly is necessary to replace Winograd's MICROPLANNEft code; exploratory development has been done using QA4, a procedure-oriented programming system particularly suited for work in artificial intelligence because of its flexibility and special features (see Rulifson, Derksen, & Waldlnger, 1972). With revisions to the parser in QA4; new techniques are necessary to facilitate the accommodation of semantic and pragmatic Information and to simplify the dynamic reordering of paths through the grammar.

At various times up to this point, prosodic information has been mentioned. Knowledge about stress, intonation, and pauses should be valuable in any speech understanding system, but it assumes special importance in an approach like ours. Prosodic data perform some of the functions for spoken language that punctuation does for text. Intonation contours can suggest sentence type—question, command, statement; together with stress and pause they may help identifying clause and phrase boundaries and signal parts of an utterance with particular semantic import (see o'Malley, 1973). Our system requires contexts (of various kinds) for useful prediction. Prosodic information may provide guidance at the beginning of a dialog or even of an utterance when other kinds of constraints are less effective.

Acoustic Processing

The speech data currently used in the SRI system are obtained in a quiet room using a BX-K433 condenser microphone and an Ampex AG 500 tape recorder. An analog tape is produced at 7-1/2 inches per second recording speed. The speech data on the tape arc then digitized in segments of up to 3.1 seconds in length. A presampling low-pass filter with an 8-kHz bandwidth is employed to reduce aliasing errors, and the digitization is accomplished by a 12-bit A/D converter operating at a rate of 20,000 samples per second.

The raw digital data are processed further by digital filtering. Five rms values (root-mean-square, an energy measure) of the time series data are calculated in each 10-millisecond interval of time. The first provides an amplitude value for the unfiltered time series. The other four values are from digital' filters with bandpass characteristics of 80-200 Hz, 300-1000 Hz, 500-2800 Hz, and 3.2-6.8 kHz. A linear predictive coding (LPC) analysis, using an algorithm developed by Markel (1971), provides formant frequencies and amplitudes by finding peaks in a 128-point spectrum.

In the first version of the system the strings of rms values were used in an algorithm that classified each 10-millisecond time segment as one of six events: silence, unvoiced turbulence, voiced turbulence, voiced stop, vowel-like, none of the preceding. The filter outputs and preliminary classifications of each segment are stored in disk files together with the formant frequency and amplitude data from the LPC analysis.

Word Verification

Procedures for word verification relate the words predicted by syntactic and semantic processing to the acoustic data. The input to the word verifier from pintle was a set of words that could be expected to occupy the next position in the utterance. The result of word verification is a subset, possibly empty, of the candidate words ordered according to degree of agreement with the acoustic data at that location in the utterance.

Since Pintle was written in LISP and the acoustic processing is done in FORTRAN, it was necessary to develop procedures for communication between the two languages. An interface package makes it possible for a LISP program to create a fork (an independent process in the time-sharing system) containing a FORTRAN program, to share directly accessible data with that program, and to call functions in that program according to standard FORTRAN conventions.

For each candidate word, there is a function that tests for that particular word. The correspondence between the expected form specified in the function and the contents of the acoustic stream is expressed as one of four confidence levels: positive, possible, unlikely, and impossible. For the first three levels, the function also returns an estimate of the ending position of the word in the acoustic stream.

The word verifier collects the results for each word in a set, eliminates the impossible words, and constructs a list ordering the rest of the words according to confidence level. The word with the highest ranking is returned to Pintle; any others are saved on a backup list to be used successively if their predecessor does not lead to the prediction of a new set of words, one or more of which can be found in the utterance. The ending position of the accepted word is used as the starting point for testing words in this new set.

To illustrate the word verification procedure, consider the word BOX in the sample sentence. It was one of the words predicted by Pintle at a location beginning approximately 1.34 seconds after the beginning of the utterance. The word function for BOX produced the following actions:

1. Increment the time pointer by 170 milliseconds.
2. Attempt to find a vowel-like string in a 200-milllsecond window centered at the incremented time pointer.
3. If Stop 2 is successful:
   (a) Search for a voiced stop ahead of the vowel-like string.
   (b) Search for silence at the end of the vowel-like string. If a silence is found, search for unvoiced turbulence after the silence. Return a confidence level, where appropriate, for each search.

4. Examine the vowel-like string as follows:

   (a) Calculate the average frequencies of the first and second formants.

   (b) Calculate the average slope of the first and second formants.

   (c) Look for discontinuities in the first and second formants.

   If there are significant discontinuities or rapid changes in formant frequencies, return the value impossible.

5. Combine the results of the consonant search from Step 3 and the analysis of the vowel-like string in Step 4 as follows:

   (a) If the average formant frequencies are reasonable for the vowel [a] and all consonant searches are successful, return positive.

   (b) If the average formant frequencies are reasonable, but a consonant search failed, return possible,

   (c) If the average formant frequencies are unreasonable but all consonant searches are successful, return unlikely.

   (d) If the average formant frequencies are unreasonable and at least one consonant search failed, return impossible.

In the example, the confidence level for BOX was positive. The results showed a vowel-like string with first and second formant values consistent with ta] in the interval 1420 to 1600, a voiced stop before the vowel-like string in the interval 1340 to 1410, silence after the vowel-like string from 1650 to 1690, and unvoiced turbulence from 1690 to 1910.

It should be clear that a word verification procedure of this kind was designed for use in a system with powerful syntactic and semantic constraints. In the analysis of the second noun group in the sample sentence, before BOX was confirmed, a set of adjectives was processed by the word verifier. All of the words were rejected except BLACK, for which the confidence level was unlikely. Pintle accepted BLACK tentatively, but that would have had to be the end of the sentence, and PUT THE BLACK BLOCK IN THE BLACK is syntactically and semantically unacceptable in the current system. Consequently, Pintle backtracked and looked for nouns. Of the set predicted, BOX was confirmed with the highest confidence level. If BLOCKS had been a member of that set, the word verifier might have returned positive as well. However, since things cannot be put in blocks, that word would be excluded, on semantic groundB, from the set to be considered.

It is obvious that the word verification procedures in this form would not allow subtle discriminations. However, the addition of more complex and powerful acoustic/phonetic rules to the analysis and decision-making parts of each word function should permit significant expansions of the system capabilities. The word verifier strategy is particularly appropriate for our system design, since the syntactic and semantic decisions involve words rather than phonemes, allophones, or other phone-like units. Furthermore, the word verifier provides a way to deal with a significant subset of coarticulation problems that would be quite troublesome in a phoneme-verifier approach.

To a considerable extent, changes in the word verification procedures depend directly on increasing sophistication in acoustic processing. However, as indicated in the previous section, the need is not for new techniques for acoustic analysis but rather for ways to extract more information from the data we have. Furthermore, we believe that the motivation for changes should come primarily from the requirements of word verification. Our current efforts are directed toward providing more subroutines for acoustic parameterization in order to refine the initial classification provided by the digital filter analysis and to provide additional formant data. For example, detectors have been added that allow fricatives *to* be distinguished reliably, *so* that s, sh, f-th, and their voiced counterparts now are classified. We also have developed vowel segmentation and classification procedures that extract boundaries within vowel-like strings, smooth formant curves, and plot slopes and standard deviations of formants. Our goal is to provide a variety of general procedures that can be used in the preparation of word functions for use in word verification.

DISCUSSION

The major focus in this paper has been on the role of syntactic and semantic analysis in our speech understanding system--and appropriately so for presentation at a conference on artificial intelligence. But in describing the first version of the system, we have emphasized the inadequacies of that implementation and the requirements we see as necessary in succeeding versions. Probably the major result of this early work is the design of a parser that we believe can accommodate all of the sources of knowledge required for understanding speech. It is described in detail in the paper by Paxton and Robinson <1973) presented at this Conference. Such a parser would be well suited to the word verification procedures and the kind of acoustic processing performed in our system.

Another result of our first year is an appreciation of the interdependence of the task domain and the system design. A system like ours, which stresses syntax and semantics, is particularly appropriate for conversations with a person that involve some relatively complex task and that require a sequence of interactions. Dynamic changes in the situation reflecting progress toward some goal can provide the kind of semantic constraints that will improve the accuracy of its predictions. The "blocks world" is relatively shallow; it does not easily accommodate dialogs of the kind desired. Consequently, we have changed our task domain to the assembly and repair of small appliances, beginning with a leaky faucet. In the course of modelling this world and establishing the knowledge necessary lor its understanding, we will be addressing problems central to the whole field of artificial intelligence.

Our project can benefit from any work in artificial intelligence that illuminates semantics, pragmatics, and the process of representation. Of course, we will not reject insights that reflect on acoustics, phonetics, prosodies, and ancillary aspects of linguistics. We do need to address all of these facets for a successful system., but we certainly cannot expect to solve all of the problems ourselves.

ACKNOWLEDGEMENTS

The following people have been involved in the research on speech understanding at SRI reflected in this paper: Sharon Baranofsky, Dick Becker, Steve Coles, Earl Craighill, Mike Hecker, Grant Hoyt, Bob McPartland, Bill Paxton, Tito Poza, Ann Robinson, Jeff Rulifson, Jim Young. We are also indebted to other participants in the ARPA Program on Speech Understanding for the perspectives they have provided.

REFERENCES

1. Barnett, Jeffrey. "A Vocal Data Management System." In: Air Force Cambridge Research Laboratories. 1972 Conference on Speech Communication and Processing, Newton, Massachusetts, April 24-26, 1972. Air Force Cambridge Research Laboratories, Bedford, Massachusetts, 22 February 1972. (AFCRL-73-0120, Special Reports No. 131) pp. 340-343.

2. Enuan, Lee D.; Fennell, Rick D.; Lesser, Victor R,; Reddy, D. Raj,"System Organizations for Speech Understanding." In: Third International Joint Conference on Artificial Intelligence. Stanford California, 20-23 August 1973. Advance Papers of the Conference. Stanford Research Institute, Menlo Park, California, 1973.

3. Forgie, James W. Speech. Semiannual Technical Summary. Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts, 31 May 1972, 15 pp. (ESD-TR-72-165).

4. Forgie, James W. Speech. Semiannual Technical Summary. Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts, 30 November 1972, 17 pp. (ESD-TR-72-263).

5. Hill, David R. "Man-Machine Interaction Using Speech." In: Yovits, Marshall C. ed. Advances in Computers. Volume 11. Academic Press, New York - London, 1971. pp. 165-230.

6. Lea, Wayne A. "Computer Recognition of Speech." In: Current Trends in Linguistics, Volume 12. Edited by Thomas A. Sebeok. Mouton & Co., The Hauge, The Netherlands, 1972, pp. 1561-1620,

Markel, J. D., "Format Trajectory Estimation from a Linear Least-Squares Inverse Filter Formulation," SCRL Monograph No. 7, Speech Communication Research Laboratory, Santa Barbara, California (October 1971),

Newell, A., et al., "Speech Understanding Systems: Final Report of a Study Group," Carnegie-Mellon University, Pittsburgh, Pennsylvania (May 1971). To be published by North-Holland Publishing Company, Amsterdam, Netherlands, 1973.

O'Malley, Michael H. "The Use of Prosodic Units in Syntactic Decoding." Presented at the Session on Linguistic Units at the 85th Meeting of the Acoustical Society of America, Boston, 13 April 1973 (University of Michigan, Ann Arbor, Michigan, 1973).

10, Paxton, William H.; Robinson, Ann E, "A Parser for a Speech Understanding System," In: Third International Joint Conference on Artificial Intelligence, Stanford, California, 20-23 August 1973, Advance Papers of the Conference. Stanford Research Institute, Menlo Park, California, 1973.

11. Reddy, D. Raj; Erman, Lee D.; Fennell, Rick D.; Neely, Richard, "The Hearsay Speech Understanding System." In: Third International Joint Conference on Artificial Intelligence, Stanford, California, 20-23 August 1973. Advance Papers of the Conference;, Stanford Research Institute, Menlo Park, California, 1973.

12. Rulifson, J. F., Derksen, J. A., and Waldinger, R. J., "QA4: A Procedural Calculus for Intuitive Reasoning," Technical Note 73, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (November 1972) .

13. Walker, D. E., "Speech Understanding Research," Annual Report, Project 1526, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (February 1973).

14. Winograd, T., "Procedures as Representation for Data in a Computer Program for Understanding Natural Language," Report MAC-TR-84, MIT Project MAC, Massachusetts Institute of Technology, Cambridge, Massachusetts (February 1971). Published as Understanding Natural Language (Academic Press, New York, New York, 1972).

15. Woods, William A.; Makhoul, John. "Mechanical Inference Problems in Continuous Speech Understanding," In: Third International Joint Conference on Artificial Intelligence, Stanford, California, 20-23 August 1973, Advance Papers of the Conference, Stanford Research Institute, Menlo Park, California, 1973.