MECHANICAL INFERENCE PROBLEMS IN CONTINUOUS SPEECH UNDERSTANDING

W. A. Woods, J. Makhoul

Bolt Beranek and Newman Inc.
Cambridge, Massachusetts 02138

Abstract

This paper presents and discusses examples of mechanical inference problems which must be solved in order to construct effective mechanical speech understanding systems. The examples are taken from incremental simulations of a prototype speech understanding system which will use syntactic, semantic, and pragmatic information as well as acoustical and phonological information to mechanically "understand" continuous speech utterances.

Introduction

In experiments in spectrogram reading [1] the performance obtained by human experts for phonetic segmentation and labeling without conscious appeal to syntactic, semantic, or vocabulary constraints was: approximately 75% of the segments correctly labeled (with either a complete or a partial phonetic specification), 15% mislabeled, and 10% segments missed. The fact that human experts with years of experience in looking at spectrograms and a detailed understanding of the acoustic characteristics of speech sounds find it impossible to uniquely decide which of several possible phonemes are present in a given segment of speech signal, and the fact that they make a significant number of errors in both segmenting the signal into phonetic units and in the labeling of these units, make it unlikely that any mechanical acoustical processing component will be able to segment and label continuous speech signals with very high reliability using only acoustic information. Moreover, it is likely that this indeterminacy in the acoustic domain is a fundamental property of human speech and not just an inadequacy in the analyzer.

However, in the same experiments, when the spectrogram reader used syntactic, semantic, and vocabulary constraints to attempt to identify the words in the sentences (using a computerized word retrieval routine which facilitated the vocabulary searches) the success rate for word identification was 963. There is hope therefore that with the proper use of syntactic, semantic, and vocabulary constraints one could build a system to understand continuous speech at a comparable level even though the acoustic segmenter and labeler operates with a significant error rate. Of course, in both the initial segmentation and labeling and in the subsequent application of syntactic and semantic constraints, the attainment with a mechanical algorithm of performance comparable to that of a human is no small task.

The BBN Speech Project

The speech project at Bolt Beranek and Newman [2,5,6] is endeavoring to construct a computer system which approaches the performance of human spectrogram readers at deciphering the meaning of continuous spoken sentences. The task of this system will be to "understand" spoken sentences and take appropriate actions. Note that this task does not include producing an accurate phonetic transcription of the input or even necessarily an accurate list of the successive words of the input (although it would be hard to imagine it getting the appropriate action if it did not in fact identify most of the words). What we are emphasizing here is that in a situation in which the acoustics is unable to resolve the decision between two phonemes or between two words at some point in the sentence, but the remaining components are able to decide the meaning of the sentence in any case (e.g. the meaning is the same regardless of which phoneme or word is chosen), then the sentence will be deemed to have been correctly understood. It is this difference between what is required for a correct output that distinguishes what the members of the ARPA speech project [3] have been calling "speech understanding" from the more traditional "speech recognition".

By examining the teletype protocols of the Klatt and Stevens experiment [1], we were able to gather considerable information about the problem solving processes and strategies which those researchers used to untangle the meanings of spectrograms. On the basis of these protocols one can conceptually decompose the speech understanding process into a number of components or routines corresponding to different types of knowledge and inference techniques applied. These components included (1) EXTRACT, the routine which performs the phonetic segmentation and labeling of the acoustic signal (both segmenting and labeling are intimately cross connected), (2) LEXRET, a lexical retrieval routine which recovers possible words from the vocabulary on the basis of partial phonetic information (this component was machine implemented in the Klatt and Stevens experiment), (3) MATCH, a routine which compares a given candidate word against the speech signal at a given point and determines the quality of the match (this component is intended to include the use of phonological and acoustic-phonetic rules for

characterizing the changes which phonemes may undergo *in specific* sentential environments*)*, (4) SYNTAX, a component which makes judgements of syntactic acceptability of sequences of words and may also propose words on the basis of syntactic context {this component may eventually also correlate the prosodies of the speech signal with the syntactic structure of the sentence), (5) SEMANTICS, a component for judging the semantic acceptability of a partial utterance and for proposing words semantically motivated by context, and (6) PRAGMATICS, a component which encompasses that knowledge which one has about the immediate context of the dialog that is not part of his general syntactic and semantic information (such information includes knowledge concerning the user, the user's state, the context of the dialogue, etc.). All of these six components are knit together by some governing problem solving strategy which we will identify as a separate component and call CONTROL.

Incremental Simulation

In the Klatt and Stevens experiment [1], the LEXRET component was implemented in a computer and all of the other components resided in the head of the human spectrogram reader. The teletype protocol, which constitutes a record of the information exchange between the LEXRET component and CONTROL, was very informative but left many questions unanswered. For example, it was difficult to tell where in the spectrogram the person was looking (one had to deduce it from the phonetic information that he was giving LEXRET), and one could not tell when and where the experimenter was performing MATCH with words that were not returned by LEXRET (for example, small function words were almost always proposed and matched without using LEXRET). This suggested a design methodology which we have been using to gather information about speech understanding problems and to construct a prototype system. The method, which we have been calling "incremental simulation", consists of filling the roles of the different components of the system partially with humans and partially with computer programs and attempting to understand spectrograms while keeping protocols of the information exchanged between components.

A human filling the role of the CONTROL component, for example, would be attempting to devise a strategy to use the information from the other components to effectively "understand" spectrograms. As he finds certain functions that he performs becoming mechanical and boring, he writes computer programs to perform them and in this way gradually replaces himself with a computer program. He may then monitor the behavior of this component and make modifications as he sees fit. Humans filling the roles of other components perform the same functions simultaneously attempting to help the overall understanding process by giving the best information they can, trying to formulate algorithms which will generate similar performance, and simulating these algorithms to assess their performance.

The incremental simulation approach has the advantage that one can quickly obtain a feeling for some of the difficult problems without having to wait for a complex overall system to be built and then discovering a fatal flaw in the system design. In this paper, we would like to share with you some of the examples of inference problems which we have encountered as the result of such simulations.

Restricting the Scope

Both for the initial phonetic analysis and for the subsequent linguistic processing, matching the performance of a human in the task of speech understanding requires a great deal of specific knowledge -- knowledge of the behavior of speech sounds, of syntactic constructions and of the semantic relationships between words. Human beings spend something like the first six years of their lives approximately one half time at the task of learning this information, and the updating and refining of it goes on all their lives. Moreover, this learning is mostly at a subconscious level and one is not aware of what he "knows" in these areas. For example, any layman can tell you when a sample of synthesized speech sounds unnatural, but he cannot (and in many cases a linguist or speech scientist cannot either) tell you what rule or regularity of speech is being violated.

Since the present state of knowledge in the areas of acoustic-phonetics, phonology, syntax, and semantics is far from complete, it is necessary to restrict our scope to some extent in order to limit the amount of knowledge that we require. In the areas of syntax and semantics, language understanding projects have achieved such limitations by restricting attention to particular data bases about which questions will be asked. This results in a restricted vocabulary, a restricted set of meanings for the words and some small restriction on the range of English constructions that one might use. In the BBN Speech Project we have achieved such a restriction by focusing on an existing natural language question-answering system in which the syntax and semantics have already been formalized — the Lunar Sciences Natural Language Information System [4] hereafter referred to as LUNAR. This is a system in which a lunar geologist can type English questions such as "What is the average concentration of rubidium in high-alkali rocks?" and receive an answer computed from a data base of chemical analyses for the Apollo 11 lunar rock samples. The choice of this system as a vehicle for speech understanding research has a number of advantages aside from the fact that the system already existed. Among other things it contains a large and interesting vocabulary and an extensive grammar.

A Sample Simulation

To give you a flavor of what an incremental simulation involves and also to begin our exposition of inference problems encountered, let us follow through the steps of an abbreviated *version* of a simulation. In this simulation, EXTRACT has been manually done off-line, LEXRET and a very crude MATCH component are implemented in the machine, and SYNTAX, SEMANTICS, PRAGMATICS, and CONTROL

reside in the head of the human simulator. In addition there is a machine implemented BOOKKEEPING component which can be used to keep track of what has been done and what has been discovered as the simulation progresses. For the sake of brevity, we will not follow out all of the blind alleys and extraneous processing which was done in the original simulation nor will we give all the details of the computer output.

A fragment of the off-line simulation of EXTRACT is shown in Figure 1. It consists of a sequence of partial phonetic descriptions (some of which may be optional as indicated, meaning that there may or may not be a segment of the indicated type). Partial phonetic descriptions such as (OR L W) and (AND -VOICED PLOSIVE) give the system a mechanism for dealing with ambiguities or indeterminacy in acoustic feature detection. Similarly, the possibilitiy of optional segments provides a way of dealing with ambiguities of

```
 0   (OR L W)
 1   FRONTV
 2   (OR S Z)
 3   (AND -VOICED PLOSIVE)
 4   (OR (AND -VOICED PLOSIVE) DH)
 5   (AND FRONTV (NOT IY))
 6   (AND -VOICED PLOSIVE)
 7   (OPTIONAL S)
 8   (AND FRONTV -HIGH)
 9   (OR S Z)
10   (AND FRONT -HIGH)
11   (OPTIONAL EY EH AE AX)
12   M



50   L
51   (AND -HIGH (NOT ER))
52   (OR K G)
53   (AND -HIGH BACK)
54   (OPTIONAL {OR L W))
55   (OPTIONAL VOICED)
56   (END OF SENTENCE)
```

Figure 1. Phonetic Transcription from Spectrogram

```
 0   (L W)
 1   (IY IH EY EH AE AX)
 2   (S Z)
 3   (P T K CH)
 4   (P T K CH DH)
 5   (IH EY EH AE AX)
 6   (P T K CH)
 7   (OPT S)
 8   (EY EH AE AX)
 9   (S Z)
10   (EY EH AE AX)
11   (OPT EY EH AE AX)
12   (M)



50   (L)
51   (EY EH OW AH AX AE AA AO)
52   (K G)
53   (OW AH AX AA AO ER)
54   (OPT L W)
55   (OPT B D G V DH Z ZH JH M N NX L R)
56   (END OF SENTENCE)
```

Figure 2. Sequence of Alternative Phonemes

segmentation. The reduction of each partial description to a list of the phonemes which could satisfy it is shown in Figure 2, and Figure 3 gives a list of the computer representations of the phonemes used in these simulations. Following through the steps of the simulation will be more effective if we withhold the identity of the sentence until the end.

The functions which will be used for this simulation are as followst

SX(n m) picks up a sequence of m successive segment descriptions beginning at position n from the output of EXTRACT. Each segment description consists of a partial phonetic description, a "confidence* figure (« 100 throughout this simulation), and a pointer to the position of the next segment. As a side effect, SX sets global variables to remember its output and the position n.

RX() calls the lexical retrieval component for words which match the pattern returned by the last call to SX starting from the beginning of the word.

MX(word n) or MX(word) matches the indicated word against the "waveform" (actually, in the simulation, against the output of EXTRACT) at position n. If n is not specified, the match occurs at the position of the last SX. MX uses a phonetic similarity matrix to evaluate closeness of match (on a scale in which 100 is an exact match) and returns a list of such numbers for each phoneme in the word. This list is terminated with a pointer to the position of the end of the word — i.e. where the next word should begin.

ADDWORD(n word c e) adds a word match to the bookkeeping table LEXTABLE beginning at position n and ending at position e with "confidence" c.

R2X() is like RX, but retrieves two-word sequences as well as single words.

RIX() is like RX, but retrieves words which contain the pattern anywhere within them — i.e. the match is not anchored to the beginning of the word as in RX.

PHONEME	SYMBOL,	EXAMPLE	PHONEME	SYMBOL.	EXAMPLE
i	IY	beat	m	M	met
I	IH	bit	n	N	net
e	EY	bait	0	NX	sing
e	EII	beT	P	P	Een
a	AE		t	T	ten
a	AA	bar	k	K	Kit
A	AH	but	b	B	bet
0	AO	bought	d	D	debt
O	OW	boat	g	G	
U	UH	bush	h	HI!	hat
u	UW	boot	f	F	Tat
e	AX	aEout	ft	TH	thing
t	ER	bird	s	S	sat
aw	AW	down	s	SI1	shut
ay	AY	buy	v	V	veil
oy	OY	boy.	o	DH	that
y	Y	you	z	Z	zoo
w	W	wind	1	ZH	measure
r	R	rent	c	CH	church
1	L	let	3	JH	Judge

Figure 3. Computer Phonetic Representations

Simulation

To begin our simulation we extract a sequence of partial phonetic descriptions of length 3 at the beginning of the sentence by calling SX and we then retrieve words matching this sequence by calling RX. The computer protocol is:

```
43←SX(0 3)
1: ((OR L W) 100 1)
    (FRONTV 100 2)
    ((OR S Z) 100 3)
44←RX()
LESS
LIST
WAS
```

We get back three words, •less", "list", "was", of which the second two sound good to us (pragmatically?) as words which might start a sentence in the lunar geology application. The word "less" is syntactically possible for a declarative sentence, but the sentences that geologists are expected to say to LUNAR are either questions or imperatives. He proceed to match the two favored words for acoustic match quality by calling MX, and add the two words to LEXTABLE by calling ADDWORD:

```
45←MX(LIST)
1: (100 100 100 100 . 4)

46←MX(WAS)
1: (100 53 100 . 3)
2: (100 100 100 . 3)
47←ADDWORD(0 LIST 100 4)
(0 (LIST 100 4))
48←ADDWORD(0 WAS 100 3)
(0 (LIST 100 4) (WAS 100 3))
```

The two matches for "was" are due to two different phonetic spellings for the word in the dictionary.

Between the two words "list" and "was" we favor the first for a number of reasons ™ it is a longer match and therefore less likely to be an accident and "list" is one of the most frequent words that begin sentences in the lunar geology application (along with "give" and "what"). Also, the geologist is less likely to ask a question in the past tense. We therefore begin working under the assumption that the first word is "list", and we look for words that start at 4 next:

```
51←SX(4 3)
1: ((OR (AND -VOICED PLOSIVE) DH)
                100 5)
    ((AND FRONTV (NOT IY)) 100 6)
    ((AND -VOICED PLOSIVE) 100 7)
52←RX()
PAPER
POTASSIUM
TITANIUM
THAT
53←MX(POTASSIUM)
1: (100 100 100 40 72 62 100 73 . 12)
2: (100 100 100 40 72 62 100 100 . 13)
3: (100 100 100 100 100 78 100 100 . 13)
4: (100 100 100 100 100 78 73 69 . 14)

54←ADDWORD(4 POTASSIUM 78 13)
(4 (POTASSIUM 78 13))

55←MX(TITANIUM)
1: (100 100 100 43 77 62 100 73 . 12)
2: (100 100 100 43 77 62 100 100 . 13)
3: (100 100 100 100 64 78 100 100 . 13)
4: (100 100 100 100 64 78 73 69 . 14)
5: (100 53 63 70 100 64 78 100 100 . 13)
6: (100 53 63 70 100 64 78 73 69 . 14)
7: (100 53 63 78 47 77 78 73 69 . 14)
8: (100 53 63 78 47 77 66 76 87 . 15)
```

The multiple matches here are due to different phonetic spellings and to the alternatives of skipping or not skipping optional segments in the output from EXTRACT. The third match of "potassium" is pretty good and potassium is also good semantically. None of the "titanium" matches are inspiring. In the original simulation we also matched "paper" and "that". The "paper" match was not too bad and the "that" match was exact, but the "potassium" match was such a long one and so good semantically that it was preferred.

Since we are working on the assumption that the first word is "list", we expect the beginning of a noun phrase after it • and therefore English determiners are likely words to occur. At this point, the syntactic component is capable of predicting determiners as possible next words and so we try a match of the syntactically proposed word "the". This word was not retrieved by RX (as "that" was) since it contains fewer than 3 phonemes. The small function words (such as "the") are the most ambiguous words to recognize since they are so short that the probability of accidental match is high and also because they are seldom stressed and are usually very much reduced in their pronunciation. The ability of the syntactic component to predict the places where they might occur is an important source of information to tap. The word "the" matches and is added to LEXTABLE, but the "potassium" match is favored and we pursue that alternative. (Here and elsewhere we will omit the actual computer printout for brevity.)

Since the "potassium" match ended at 13, we begin looking for the next word at 13 and find only "rubidium" which matches perfectly as follows (two phonetic spellings both match):

```
60-MX (RUBIDIUM)
1: (100 100 100 100 100 100 100 100 . 21)
2: (100 100 100 100 100 100 100 100 . 21)
```

The perfection and the uniqueness of this match convince us that we are on the right track and confirm our belief that the previous word was "potassium" (and not "the" for example). One version of this simulation was done in front of an audience of kibitzers who at this point were trying to figure out how one could syntactically have two words such as "potassium" and "rubidium" in a row. They concluded that it might be a conjoined list of the form "A, B, and C" and so they proposed (syntactically?) the word "and" at this point. The match unfortunately was unsuccessful.

For those familiar with the LUNAR system, this pair of words together suggested an entirely different next word (semantically!) since the potassium/rubidium ratio is one of the standard correlates of age for the lunar

samples. Thus the next word could have been predicted semantically, although in this simulation it was discovered by LEXRET with the following match:

```
65←MX(RATIO)
1: (71 53 73 56 100 . 26)
2: (100 100 100 100 82 . 27)
3: (100 100 100 100 41 . 28)
```

Whenever one has just recognized a noun or a verb which can undergo regular inflection by suffixation, it is appropriate to look for such suffixes. In this case, since there was no determiner on the noun phrase , the syntactic component should predict that the noun be plural. We successfully match "-S" at position *27* and we add "ratios" to LEXTABLE from 21 to 28.

Again, we are now in a context where the syntactic component can predict small function words — in this case prepositions modifying the noun. Also, from semantics, we know that one computes potassium/rubidium ratios in (or for or of, etc.) samples. Thus syntax can predict a preposition, and given this, semantics can predict which ones. One way to take advantage of this is to call SX for pattern sequences of length 2 and then scan the results of RX for small prepositions. This retrieves the word "for" with the following matchest

```
71←MX(FOR)
1: (100 82 50 . 31)
2: (77 38 76 . 32)
3: (100 100 50 . 31)
4: (77 72 76 . 32)
5: (100 89 . 30)
6: (77 41 . 31)
```

The word "for" satisfies our prediction well but the match quality (the fifth one is best) is not especially great. We therefore follow out our prediction a little further (before adding "for" to the table) by predicting (semantically!) the word "sample":

```
72←MX(SAMPLE 30)
1: (100 100 100 100 100 100 . 36)
2: (100 100 100 100 100 56 . 37)
3: (100 100 75 74 100 56 . 37)
4: (100 100 75 74 77 72 . 38)
```

The perfection of this match confirms our hypothesis and we add both "for" and "sample" to LEXTABLE. Again we check for plural endings and again syntax could predict a plural from the absence of a determiner. This results in adding "samples" to LEXTABLE from 30 to 37.

The sentence now seems to read "List potassium/rubidium ratios for samples ...", and we are now looking at position 37 where we find the words "data", "that", and "not". The word "data" looks impossible and "not" looks unlikely, but "that" looks very good (syntactically!) as the beginning of a relative clause. We find that it matches ending at 40, so we add it to LEXTABLE and begin at position 40 where we find a perfect match for "contain" ending at 46 (among 4 words returned by RX). The verb "contain" looks good as the verb of the relative clause, especially since semantics knows that samples can contain minerals and elements, etc. Its match quality is excellent, and in our enthusiasm for the current path we don't even look at the others.

At this point in one simulation, a member of the audience who had had some experience with the LUNAR system took a wild stab and predicted (pragmatically!) the word "olivine", the name of a mineral. The match however was unsuccessful. We begin looking for the next word in the usual way but find no words retrieved by RX. We try again with R2X (one way of finding shorter words) with the following result:

```
80←R2X()
FIRST-WORD

IN
-ED
-ING
AN
ON

SECOND-WORD
16 WORDS,  GO AHEAD? NO
```

Aha! The verb is inflected, so we try the two inflectional endings, both of which match ending at 48. The verb must then be either "contained" or "containing". The ending "-ed" is consistent with our current hypothesis and we look now at position 48:

```
93←SX(48 3)
1: (S 100 49)
    ((AND -HIGH (NOT EY ER)) 100 50)
    (L 100 51)
94←RX()
SULFIDE
95←MX(SULFIDE)
1: (100 100 100 77 17 85 64 . 55)
2: (100 100 100 77 17 85 100 . 56)
```

Sulfide, which started out right is obviously not the right word, but there are no other words retrieved. The match of everything else up to this point has been very good, so we are reluctant to back up to other possibilities. It is possible that there is an acoustic labeling error in these first three segments. One way to check is to move right and start doing "unanchored" matches using RIX. (Notice that semantics can tell us to look for a mineral, a chemical element, an isotope, etc.) We find nothing particular using 3 segments at 49, but using 3 segments at *50* we get:

```
99←SX(50 3)
1: (L 100 51)
    ((AND -HIGH (NOT ER)) 100 52)
    ((OR K G) 100 53)
100←RIX()
PUBLICATION
SILICON
101←MX(SILICON 48)
1: (100 77 100 100 100 100 83 . 55)
2: (100 77 100 100 100 100 100 . 56)
```

Silicon fits beautifully and shows us that the *second* segment was mislabeled. Also we are at the end of the sentence, so we print out LEXTABLE (in real simulations one does this often to see what he has found out so far):

```
103←(PRINTDEF (SORT LEXTABLE T))
  ((0 (LIST 100 4)
      (WAS 100 3)
      (LESS 100 3))
   (4 (POTASSIUM 78 13)
      (THE 100 6))
   (13 (RUBIDIUM 100 21))
   (21 (RATIO 82 27)
       (RATIOS 82 28))
   (28 (FOR 89 30))
   (30 (SAMPLE 100 36)
       (SAMPLES 100 37))
   (37 (THAT 100 40))
   (40 (CONTAIN 100 46))
   (46 (-ING 100 48)
       (-ED 100 48))
   (48 (SILICON 77 56)))
```

It looks like the sentence is "List potassium/rubidium ratios for samples that contained silicon". Everybody in the audience was happy. However, that was not the correct analysis, and there are several morals to be gleaned from this example. Recall that at position 37 the word "that" looked so good as the beginning of a relative clause that we didn*t even match "data" or "not". At that point syntax could have told us that "not" could begin a reduced relative clause, especially if the next word were the -ing form of a verb, but reduced relative clauses are relatively rare and LUNAR's grammar postpones looking for them until it has tried other things. When one simulator first analyzed this sentence, he associated all of this information with the rejected word "not" and when the inflection "-ing" occurred at position 46, he revised his opinion of "not" and made it an equal competitor with "that". The analog for a computer program would be to suspend a process with a "demon" looking for an "-ing" verb.

When we look, we find that "not" matches perfectly ending at 40, so another possible reading for the sentence is "List potassium/rubidium ratios for samples not containing silicon". We have to decide between these two alternatives. Note that in this example we happen to have two competing interpretations with exactly opposite meanings!

It turns out that there are a number of grounds on which one can base the choice between these two readings. They illustrate the kinds of redundancy that are available to resolve such ambiguities if we have the appropriate inference devices. First, pragmatically, it is unlikely that a geologist talking to LUNAR would have referred to a sample containing silicon in the past tense unless both he and the system had reason to believe that the sample no longer existed (and the data base of LUNAR doesn't know about such things). In fact, the same member of the audience who guessed olivine at position 48 raised this objection to the first analysis before it was pointed out that the second analysis was possible. So for pragmatic reasons alone we would favor the second analysis (even enough to go looking for it when it had not yet been detected).

If one had not resolved the ambiguity on pragmatic grounds, one could also have done it on phonological grounds. When combining the inflectional endings such as "-ed" with verbs such as "contain", there are phonological constraints which determine how the "-ed" ending will sound. In our phonetic dictionary "-ed" has two spellings: (D) and (AX D). In the above simulation, on closer inspection one can tell that the spelling which matched was the second, corresponding to the three syllable pronunciation "con-tain-ed" rather than the correct "con-tained". Thus, by using such phonological rules when matching proposed inflected forms, we could have ruled out "contained" in favor of "containing".

Finally, we could have resolved the ambiguity acoustically by calling a variant of the MATCH component to give relative scores to the competing word pairs "not"/"that" or "-ed"/"-ing", When the person who had simulated the off-line EXTRACT was asked whether the word at 37 looked more like "that" or "not", he said definitely "not". Thus, one could have resolved the differences by having an acoustic "word ambiguity resolver" which given two (or more) words tries to determine which is the best match. This could be done for example by refining the MATCH component and taking the word with the best match score.

Discussion

The preceding sample simulation, while it gives a good impression of some of the situations encountered in continuous speech understanding, is untypical in several respects. First, the acoustic segmentation and labeling on which this simulation is based is unusually good and the branching of alternative possibilities is unusually narrow. That is, the remarkable degree to which syntactic and semantic intuitions led us directly along the right path without extensive blind alleys in this simulation is unusual. Also, there were no segmentation errors (i.e. no missing segments). All of the questionable segments had been labeled as optional by the segmenter and therefore the LEXRET and MATCH components did not have to cope with segmentation errors. Other sentences that we have simulated have contained such errors, and more powerful word retrieval and match components have been developed to deal with them.

A situation that has occurred often in simulations is that rather than receiving a list of words to choose from as a result of a call to LEXRET, one finds no words at all. We encountered one such situation in the sample simulation due to the labeling error at position 49 and we recovered the word by doing unanchored matches further to the right. However, if the input signal had been more severely garbled so that the resulting match did not look so good or if the rest of the words in the sentence had not matched so well, we would not have been so easily able to choose between this path and the possibility that one of the previous word matches was incorrect and some other word match at some previous point might have lead to a better total match. When one finds a position at which LEXRET finds no word matches at all, then either it is a position where no words are (i.e. the previous word which ends there is an accidental match) or else there are some segmentation or labeling errors that are

blocking a word match (this of course assumes that the utterance does not contain a word that is not in the system's lexicon). In the former case, one should reject the accidental match and look elsewhere for the correct word sequence, while in the latter case, one might recall the EXTRACT component to relax or revise its previous description of that portion of the sentence or he might try a more desperate version of LEXRET and MATCH which can compensate for gross errors in the acoustics. Another possibility in the second case is to call MATCH with all of the conceivable words that could be predicted for that position by syntactic and semantic context.

In the sample simulation, we had active at every step of the analysis (until the postscript match of "not") a single "theory" (or hypothesis) about what the sequence of words in the utterance was and what the syntactic and semantic structure of the utterance was. Moreover, this theory was continually grown and refined from left to right in one continuous and unbroken development. Only in the postscript did we develop a second competing theory. This is unfortunately very unusual. The more typical situation is that there are several (or even many) competing theories developed in the course of an analysis, and some of them may be discontinuous (i.e. may relate words that are not adjacent to each other in the input, without any hypothesis for the words that fill the gap between them). In fact, when run without the incredible selectivity which the human CONTROL component can generate by hunch, intuition, or "divine guidance", *(or* perhaps just a lot of unconscious enumeration and testing), a completely mechanical speech understanding system will inevitably generate a large number of such theories which must be compared and evaluated. Without some efort to decide which of the competing theories are worth pursuing and extending, an exhaustive enumeration would quickly be swamped in the combinatorics.

A Second Example

A more typical simulation (although still easier than many) resulted in the following LEXTABLE:

```
((0 (BEEN 100 3)
    (DID 100 3)
    (DONE 100 3)
    (ANY 100 4))
 (3 (EIGHTY 100 6)
    (ANY 100 6))
 (4 (BE 100 6)
    (ME 100 6)
    (DID 90 7)
    (NEED 90 7))
 (6 (PEOPLE 100 11))
 (11 (DONE 100 14)
     (BULK 72 15))
 (14 (TEN 100 18)
     (CHEMICAL 56 22))
 (22 (ANALYSES 75 30))
 (30 (IN 100 32)
     (ON 100 32))
 (31 (MUCH 100 34))
 (32 . NONE)
 (33 . NONE)
 (34 (ROCK 100 38)))
```

This is the state of LEXTABLE after considerable searching for possible words. At this point, every segment of the input is covered by some word, so there are no obvious places in the waveform where new word matches are needed. However, there is no sequence of words which covers the entire input. Therefore there must be an acoustic error somewhere — the question is where. Starting from the beginning of the utterance, we can find the sequence of words "Did any people", which is good syntactically and aemantically, but the closest next word to the predicted verb is the past participle "done". This word is inconsistent with the preceding word sequence, but specifically it is inconsistent with "did". Prom LUNAR's transition network grammar it is possible to infer that "done" is incompatible because the grammar is predicting at this point an untensed verb. Moreover it is possible to infer from the state of the grammar that a past participle would be possible if the preciding verb was "have" or "be", and to follow back the analysis path to determine that the verb register was set by the word "did" at the beginning of the utterance. Thus, all the groundwork is present to syntactically predict either "Did any people do" or "Have any people done". It turns out that the latter was the actual utterance with the initial HH missing and the unstressed AE reduced to a schwa. Correctly analyzing this utterance requires the ability to draw the syntactic inference that the initial word was "have".

Given that we correctly discover "have", we now have a continuous sequence "Have any people done chemical analyses *on"* which is syntactically and semantieally very plausible even though the word matches for "chemical" and "analyses" are not perfect. However, there are no successful word matches beginning at position 32. Recall the general rule that the absence of word matches at a given point indicates either that the preceding word matches were accidental or that there is an acoustic error. If the possibility of an acoustic error were not considered then the first half of this rule would undo all the word matches back to position 14 (and would seem justified since the matches of "chemical" and "analyses" are not perfect). Somehow the size and semantic goodness of the current theory must keep it under consideration in spite of its acoustic flaws.

Notice that we are in a context where syntax can predict determiners. Suppose that we assume that there is a determiner here and look for the next word somewhere to the right. We find the word "rock" at 34 which makes excellent semantic sense and matches to the end of the sentence. We could now assume that there is a missing determiner from 32 to 34 with greater confidence, and if we look at the acoustics with the determination to find the best possible determiner match (however tenuous), we should come up with "this" due to a distinct acoustic "S" at 33. This is in fact the correct analysis. The unstressed "this" was pronounced something like (AX S) with the initial TH completely invisible in the spectrogram (and probably not pronounced by the speaker)•

Conclusion

We believe the two examples given in this paper convey a good picture of the types of probabilistic, plausible inferences that a continuous speech understanding system will have to be capable of making in order to extract meaning out of the speech signals that human beings produce. The task requires the integration of several inference components (CONTROL, SYNTAX, SEMANTICS, and EXTRACT) each of which has an open ended set of possible alternatives that it can pursue with smaller and smaller likelihood of success. One must have some method of dovetailing the computations of all of these components together, since any given component would effectively never finish trying increasingly remote possibilities. Moreover, it is probably essential that the individual components maintain their own data structures and special strategies tuned to the special nature of their tasks and not be subsumed under some monolithic general purpose inference procedure. Thus one of the essential tasks of CONTROL will be to balance the resource allocation among the various components in order to maximize the benefit — e.g. it would be foolish to try extremely improbable word matches when one had not yet tested the syntactic and semantic acceptability of better word match combinations.

The speech under standing problem is almost a complete microcosm of the general robot planning problem and in some ways more difficult. We have the same problems of representing "alternative worlds" (in this case our theories about the utterance), of drawing together a diversity of facts to find out about our real world environment (in this case the utterance), and of putting these facts together to produce appropriate actions. Moreover, we have the same problem (not yet effectively dealt with in robot projects) of coping with the basic uncertainty or incompleteness of the input data and the necessity to make assumptions. We have the same or even more critical need to devise inference techniques which avoid the redundant derivation of the same conclusion in exponentially many different ways, while on the other hand, we need to be able to derive the equivalent of a proof with a step missing and to use that "proof to predict the missing step.

Because of the uncertainty of the input and the open ended possibilities for error, strictly logical systematic enumeration methods for deductive inference will not suffice. The space of possibilities is too vast to search in its entirety. It is essential to have inference techniques which "play the odds" and follow out the most promising possibilities first. We must also be able to terminate and ask questions of the user when the law of diminishing returns makes that alternative more economical than continued search. We believe this to be true not only for speech understanding, but also for robot problems as well. An intelligent automaton cannot function on just those inferences which it is logically justified in making deductively — it will hardly ever have sufficient data. Rather it must constantly be making assumptions based on likelihood. However, it must know where its deductions depend on such assumptions so that it can cope with situations in which they prove false.

In the BBN speech project, we are attempting to build a system along the lines suggested here. We will be attempting to combine likelihood estimates with the inference processes that construct and refine theories and use these to control the allocation of resources among the various components. Prom this attempt, we hope not only to obtain a viable speech understanding system, but also to increase our understanding of the role of deductive inference in the face of uncertain data.

References

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

[2] Makhoul, J. and Bobrow, D.G., "Computer Assisted Recognition of Connected Speech," Workshop on Automatic Pattern Recognition of Speech, Rome, New York, September 1971.

[3] Newell, A. et al, "Speech Understanding Systems: Pinal Report of a Study Group," Computer Science Department, Carnegie-Mellon University, May 1971 (published by North-Holland / American Elsevier, 1973).

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

[51 Woods, W.A., Makhoul, J., Wolf, J. and Rovner, P., "Organizing a System for Continuous Speech Understanding," paper presented at the 85th Meeting of the Acoustical Society of America, Boston, Mass., April 1973.

(6J Woods, W.A,, Nash-Webber, B, and Bates, M., "Syntactic and Semantic Support for a Speech Understanding System," paper presented at the 85th Meeting of the Acoustical Society of America, Boston, Mass., April 1973.