A limited speech recognition system*

by DANIEL G. BOBROW and DENNIS H. KLATT†

Bolt Beranek and Newman Inc
Cambridge, Massachusetts

INTRODUCTION

A computer system which identifies words in continuous speech of an unknown speaker is beyond the current state of the art in speech recognition. LISPER, is a successful limited speech recognition system based on a set of assumptions which greatly simplify the recognition problem; within these restrictions it allows experimentation on the usefulness of a voice insertion system on the computer.

LISPER operates within limitations along a number of dimensions. Rather than use continuous speech in which segmentation is a problem, we work with messages with easily delimited beginning and termination points. The set of messages is limited in number; at any one time the vocabulary to be distinguished can contain up to about 100 items. However, an item need not be a single word, but may be any short phrase. A message list from a NASA mission context, shown in Table 1, was one of three used in testing the system.

Note that LISPER recognizes each of these messages as a unit, and does not segment a multiword utterance into individual words for recognition. The system is not designed to work well simultaneously for a number of different speakers, or achieve good recognition scores for an unknown speaker. The system is usable by any male speaker, but must be first trained by him. The training period consists of a period of closed loop operation in which the speaker says an input message, the system guesses what he says, and he responds with the correct message. In this training phase, the system will learn the idiosyncratic variations of the speaker's set of input messages. In this closed loop system, it is not unlikely that the speaker will also learn something.

*This research was supported principally by the National Aeronautics and Space Administration, and in part by the Advanced Research Projects Agency.

†Also at Massachusetts Institute of Technology, Cambridge Mass.

The LISPER system was designed as a research vehicle, as well as a pattern recognition system. For this reason, flexibility for data access and program modification were important. For this reason, LISPER was built in an extended LISP system. 1,2 This decision allowed an easy transfer during the research from a DEC PDP-1 to an SDS 940 computer.

Any pattern recognition system must have three basic components: preprocessing hardware to extract a representation of the input; programs utilizing this raw data to compute properties of the unknown input (data reduction); and a recognition or decision algorithm. Figure 1 shows a block diagram of the organization of the LISPER system. The input speech signal may be obtained from either a microphone or tape recorder. The

| one     | distance to dock |
| two     | fuel tank content |
| three   | time to sunrise  |
| four    | time to sunset   |
| five    | orbit apogee    |
| six     | orbit perigee   |
| eight   | revolution time |
| nine    | closing rate to dock |
| point   | midcourse correction time |
| plus    | micrometeoroid density |
| stop    | radiation count |
| zero    | what is attitude |
| seven   | remaining control pulses |
| minus   | alternate splashdown point |
| pressure| weather at splashdown point |
| negative| sea and wind at splashdown point |
| what is yaw | visibility at splashdown point |
| what is pitch | temperature at splashdown point |
| end repeat | skin temperature |
| affirmative | power consumption |
| inclination | fuel cell capacity |
| distance to earth | repeat at intervals |

TABLE 1—Message list from NASA context

305

From the collection of the Computer History Museum (www.computerhistory.org)
basic parameters of the input are extracted by a spectrum analyzer which consists of a preemphasis network followed by nineteen bandpass filters. Bandpass filter outputs are rectified, low-pass filtered, sampled, converted into logarithmic units, and stored on digital tape, or in the memory of the computer. The 19 filters consist of 15 filter spaced uniformly up to 3000 Hz, the range encompassed by the first three resonances of the male voice; and 4 filters covering the range from 3000-6500 Hz in which there is information about the noise component of speech. The four-pole Lerner bandpass filters each have a bandwidth of 360 Hz. An advantage of this filter system is that one does not see spectral peaks due to individual harmonics of the fundamental frequency of a male speaker; a spectral maximum in the filter outputs indicates the presence of one or more formants (resonances of the vocal tract transfer function). The spacing between filters makes resolution of individual formants poor, and we do not use formant tracking in our recognition system.

The spectral representation of a word as derived from our input system consists of 200 spectral samples, corresponding to 2 seconds of speech material and approximates the log of the short term power spectrum of the speech signal. A spectral sample consists of the outputs of the 19 bandpass filters, all sampled at a particular instant of time. A filter output, in log units, is an integer ranging from 0 to 63, covering a 45 decibel range of intensity.

The use of the logarithm of the short-time spectrum is well established as one approach to speech analysis and has often been used as the basis for recognition programs. The principal contributions of this research have been the development of a set of interesting algorithms for extracting features which characterize speech utterances, and coupling this set with a recognition algorithm capable of high quality message identification in the presence of redundant, inconsistent, or incorrect information from these properties. The recognition algorithm does not directly utilize the spectral data, nor is an intermediate phonemic transcription of the word produced. However, we have made use of the present day body of knowledge about acoustic phonetics, and the distinctive feature approach to phonological description. We discuss below some problems of these two approaches and indicate how a number of these problems have been alleviated by techniques used in LISP.

Our system has been extensively tested, and recognition scores have been obtained on three vocabularies with three different speakers. Typical of our results is attainment of recognition scores that approach 97% correct on a 54 word vocabulary after three rounds of training for a single speaker. These results compare favorably with the word recognition scores of other investigators.

Property extraction system

An unknown word is represented in the computer by a matrix of integers (filter number vs. sampled time vs. log-amplitude). The purpose of a property or feature extraction system is to reduce the information content of the input signal to a level that makes it possible for the decision or recognition algorithm to operate reasonably. The critical thing is that the information eliminated should be information which is irrelevant to the decision that the recognition algorithm must make.

The techniques used in LISP can best be appreciated by first exploring some problems involved in identifying a speech utterance on the basis of the 19 X 200 array representing a 2 second sampling of its energy spectrum. Pattern recognition schemes operating on data of this dimensionality are theoretically tenable (Sebestyen, 11), but only if repetitions of words cluster properly in the resultant vector space. It is by now well-known that speech patterns as initially stored in a computer do not cluster according to the word spoken. Some of the reasons for this are:

1. The range of intensities encountered will vary because the overall recording level is not fixed. Recording level depends on vocal effort and the distance of a speaker from the microphone.
2. An unknown word is difficult to register at the same place within 200 spectral samples because word onset time is not a simple feature to detect reliably. For example, initial voiceless fricatives may be missed, prevoiced stops are hard to treat consistently, etc. We insure that all of an utterance is contained within the two second interval by sampling the input line continuously, storing spectral samples in a computer-simulated digital tape loop, and stopping 1.8 seconds after the data first exceed a threshold.

From the collection of the Computer History Museum (www.computerhistory.org)
3. The total duration of a word is highly variable. In addition an increase in speaking rate is not manifested by a linear compression of the time dimension. Final syllables are often prolonged. Some transitions (for example, the release of a stop consonant) are not as greatly affected by changes in speaking rate as the steady-state portions of vowels. If shortened enough, vowels are likely to be reduced and consonants may not be carefully articulated, with resultant losses in spectral distinctiveness.

4. A speaker attempts to generate an utterance so that it has a particular set of perceptual attributes (or features). We do not know in detail what the acoustic correlates of these attributes are. There is a great deal of variation allowed in the acoustic properties of the signal that will still give rise to the same utterance. There is also sufficient redundancy in the message to permit a speaker to leave out certain attributes entirely. For example, the degree of stress placed on a syllable will determine the extent to which the vowel may be reduced (Lindblom, 13). Consonants in unstressed syllables may contain less frication noise, a weak stop burst release, or incomplete stop closure. Vowels may be nasalized in nasal environments. The substitution of one incomplete gesture for a consonant cluster is also common in unstressed syllables of natural speech. None of these effects would necessarily produce word recognition difficulties if they appeared consistently in the data. Unfortunately, they do not.

5. If a speaker is instructed to speak distinctly and not rapidly, some surprising and unfortunate variability in speaking habits has been experimentally detected. In an attempt to help the system, our speakers released final stops, increased the length of some syllables, and articulated unstressed syllables more carefully than they would normally. Unfortunately, our speakers appear to have found these speaking habits unnatural, and could not remember from repetition to repetition exactly what they had done to “help.” For example, final voiced stop releases gave trouble by producing short vowel segments that varied greatly in amplitude, and the words four and core were sometimes pronounced as if they had two syllables.

6. Individual speakers have vocal tracts of different sizes and shapes. It is physically impossible for two speakers to produce identical spectra for a given phone or word. A speaker makes an articulatory gesture that we, as listeners, interpret; possibly with respect to our knowledge of the spectra that he is capable of producing (Gerstman, 13). The nature and importance of the normalization process of a listener are not well understood.

7. Different speakers have articulatory habits (idiolects) that may be quite distinct. Habits include the timing and dynamics of articulatory movements and the features that a particular speaker employs to manifest a phonemic distinction. Whether recognition difficulties can be attributed to individual speech habit structures is not known. Very little quantitative data are available on the characteristics that distinguish speakers.

The variability of the spectrum of a word has led to the search for data reduction techniques to eliminate irrelevant information. An approach favored by several investigators has been to develop rules for detecting phonemes (Fry and Denes, 14; Martin et al., 15; Reddy, 16). We believe that phoneme recognition is a much more difficult problem than word recognition because it presupposes a good understanding of the cues that distinguish phonemes in arbitrary phonetic environments. For example, allophones of the phoneme /p/ may be:

1. normally aspirated, as in the word “peak” [pifik]
2. weakly aspirated, as in the word “supper” [s\'p\\'pa\']
3. non-aspirated, as in the word “spin” [spin]
4. non-released, as in the word “top” [t\'ap\']

There is no need for a word recognition program to attempt to group this disparate set of physical signals together into one phoneme. However, to the extent that algorithms for deriving a detailed phonetic-feature description of an utterance can be found, they can be of considerable help to a practical speech recognition system. Examples of feature approaches that are relevant include the work of Hughes, 17; Hemdal and Hughes, 18; Gold, 19; and Martin et al. 15.

The properties of speech which are used by the recognition program are based on the energy measures derived from the input system. The output of each filter is an elementary function of the speech input signal. We use the notation Fn(i) for the output of filter n at sample interval i; that is F1(i), F2(i), . . . , F19(i) are used for the output of filters 1 through 19 at sample interval i. The filter number n ranges from 1 for the low frequency filter to 19 for the high frequency filter; and i = 1 for the first sample interval to i = 200 for the last time sample of a two second utterance.

More complicated functions of the speech input signal can then be defined in terms of these elementary (base) functions, F1, . . . F19, in the LISP system. For
example, the following function has been found to correlate with perceived loudness, independent of vowel quality.

\[ \text{Loud}(i) = F_1(i) + F_2(i) + F_3(i) + F_4(i) + F_{12}(i) - F_7(i) \]

The output of the seventh filter is subtracted from the sum of the first four filters and filter 12 to compensate for the fact that low vowels are inherently more intense when produced with the same vocal effort. We will indicate later how a modified version of this function Loud(i) can be used to correct for differences in recording level between repetitions of the same word.

Loud(i) is useful in reducing the information that must be processed by the decision algorithm, since it helps to normalize the input with respect to a variable which is not important to recognition, namely, the recording level of the input signal. We describe in this section a number of techniques which are used to reduce the information content of the incoming signal in ways that preserve the invariance of message identity over ranges of parameters which are irrelevant to the decision about the identity of the message.

One important way we have found to reduce the information content of a function is to reduce its range of values. Thus, for example, we may define a reduced information property Amp2(i) based only on the output of filter F2.

\[ \text{Amp2}(i) = \begin{cases} +1 & \text{if } F_2(i) > 40; \\ 0 & \text{if } F_2(i) > 20; \\ -1 & \text{otherwise} \end{cases} \]

Amp2(i) is thus a 3 valued function, where we ascribe no significance to the names of the three values. As described later, a set of non-linguistic threshold properties similar to Amp2 can be used as a basis for recognition. However, we note the following problem. If the signal in F2 were varying around either of the thresholds, 20 or 40, then there would be little significance of the change of state from -1 to 0 or 0 to 1 and back; it would be much more likely due to noise than to a real variation in the input signal.

To make such properties less sensitive to noise, we introduce a hysteresis region around the thresholds to insure that a change of state is significant. A revised definition of Amp2(i) is

\[ \text{Amp2}(i) = \begin{cases} +1 & \text{if } [F_2(i) > 40] \text{ or } [F_2(i) > 37 \text{ and } F_2(i-1) = 1]; \\ 0 & \text{if } [F_2(i) > 20] \text{ or } [F_2(i) > 17 \text{ and } F_2(i-1) 
eq -1]; \\ -1 & \text{otherwise} \end{cases} \]

Most of the features we use are functions of sampled time having a range limited to a small number of distinguishable states. These functions are very sensitive to slight changes in time scale and origin; these variations in the data are irrelevant for recognition.

The time dimension is removed from a feature by transforming the time function into a sequence of transitions of states, as is illustrated in the following example:

\begin{align*}
\text{Loud} & = 1 2 3 4 5 6 7 8 9 \\
\text{Voice} & = 0 0 1 1 1 1 1 1 0 \\
\text{Voice} & = 0 1 0
\end{align*}

This transformation reduces the amount of data that must be manipulated by the program. Due to the nature of spoken language, the exact time when features change value will vary from repetition to repetition of the same word, but the essence of the word remains in the sequence of state transitions of an appropriate set of features.

No information about the word onset time, speaking rate, and speaking rhythm can be recovered from the sequence unless these parameters have an effect on the actual states that are reached. To this extent, recognition will be unperturbed by variations in word onset time, speaking rate, and speaking rhythm.

A problem that arises from collapsing the time dimension is an inability to tell whether two features were in specific states at the same time. Time removal assumes that features independently characterize a word. This is obviously false. A clear example is provided by the words "sue, zoo" which can only be distinguished by knowing whether the strident is simultaneously voiced or not. It is possible to retain some timing information by the inclusion of features that count the number of time samples between temporal landmarks in the data and change state if the count exceeds a threshold, or to include states which are entered only upon simultaneous satisfaction of two conditions.

For example, the state "-1" corresponds to voiceless segments in the definition of the spectral quality [r]-like(i). In this way we indicate in which voiced segment an [r]-like phone occurred. The sequence produced for the word "ratio" should be \(-1, 1, 0, -1, 0, -1\). A more detailed localization of the [r] is not necessary for limited vocabulary word recognition, and would certainly be more difficult. We find that very gross timing information is satisfactory for recognition.

Two features are used to make a preliminary division of a word or message into one or several segments. One feature divides a word into voiced and voiceless segments and the other divides it into syllables. Voice(i)
and Syllable(i) are then used to introduce time markers in the definitions of other features.

Features which classify the vowel quality of the stressed syllable of an utterance into one of 10 categories have been found to be very useful in recognition. The stressed syllable is identified with the aid of a function Loud(i) that computes an approximation to the perceived loudness of a vowel. These features of a stressed syllable are less likely to be affected by the natural variability that characterizes the speech process (Stevens 18).

Recognition algorithm

The recognition algorithm is a program that learns to identify words by associating the outputs of various property extractors with them. During learning, the vocabulary of words is presented a number of times, and information is accumulated about the different ways that the speaker may pronounce each word. For example, the result of the training procedure applied to the feature sequence Voice after 5 presentations of the 4-word vocabulary “one, two, subtract, multiply” might be:

<table>
<thead>
<tr>
<th>Word</th>
<th>Sequence</th>
<th>Number of times sequence occurred</th>
</tr>
</thead>
<tbody>
<tr>
<td>one</td>
<td>010</td>
<td>5</td>
</tr>
<tr>
<td>two</td>
<td>010</td>
<td>5</td>
</tr>
<tr>
<td>subtract</td>
<td>01010</td>
<td>5</td>
</tr>
<tr>
<td>multiply</td>
<td>0101010</td>
<td>3</td>
</tr>
<tr>
<td>multiply</td>
<td>01010</td>
<td>2</td>
</tr>
</tbody>
</table>

The two versions of the word “multiply” exemplify a common problem. No matter where we place threshold boundaries, there are some words that are treated inconsistently by the feature detectors.

However, our recognition algorithm is powerful enough to deal with ambiguity in the voicing assignment of some vocabulary items, and thus we need not develop a more sophisticated voicing algorithm. We made no attempt to use rate of voicing onset, offset, time between burst and energy build-up for stops, etc., as additional cues to a better feature definition. In all of the features used, the ambiguity remains at a level tolerable to the recognition algorithm. The redundancy inherent in the complete set of feature definitions allows recognition of variations of the same message. Another way of putting it is to say that there appear to be no absolute boundaries along the property dimensions we have chosen. The recognition algorithm takes this fundamental limitation into account and makes a best guess given the imperfect nature of the properties.

In the training process the program reorganizes the data for each property into a list of those sequences which have occurred. The list for Voice for our example is shown below:

<table>
<thead>
<tr>
<th>Sequence</th>
<th>(Word, frequency)</th>
</tr>
</thead>
<tbody>
<tr>
<td>010</td>
<td>one 5, two 5</td>
</tr>
<tr>
<td>01010</td>
<td>subtract 5, multiply 2</td>
</tr>
<tr>
<td>0101010</td>
<td>multiply 3</td>
</tr>
</tbody>
</table>

If a new utterance is presented to the program for recognition, and the feature Voice(i) produces the sequence 01010, then Voice will register one vote for the word “subtract” and one vote for the word “multiply.” The unknown word is identified as the vocabulary word eliciting votes from the most features. Typically, the identified word receives a vote, signifying a perfect match for that property, from only about 80 percent of the features.

In case of ties, the program makes use of information concerning the number of times a word appears at a node. Thus, if “subtract” and “multiply” tie for first place in the voting from all the features and Voice = 01010, then Voice will register 5 votes for “subtract” and 2 votes for “multiply” in the run-off between these two candidates.

The final decision procedure is an attempt to find the message from the set of possible inputs which is most similar to the current input message. Because there is a wide variation in the way people say things, the decision procedure does not insist that the current input must be like one of the prototype input strings in all ways that it was categorized; that is, it need not be suggested by all properties. In this sense, the decision procedure allows a generalization of the original training and learning by looking for a best fit without putting a bound on the goodness of this fit.

The property sequences have been considered as independent characterizations of the input message. In this case, by independent we do not mean that the computations themselves are necessarily independent, but that these descriptors of the input message are treated independently in the decision process. With the assumption of independence, the voting procedure is similar to a maximum likelihood estimate. The a priori probabilities of all messages are the same. Therefore, the most likely message is the one for which the product of the a posteriori probabilities (or the sum of their logarithms) is a maximum. For each property, every message, M_i, previously associated with the current input sequences S_i is given a (scaled) log probability of \( 100 + N_i \), where \( N_i > 1 \) is the frequency with which \( M_i \) was seen for \( S_i \). Messages not associated with \( S_i \) in training are given log-probability 0 and can therefore be ignored in the summation. These assignments of
probabilities achieve in one step a score which allows a single search to determine the most likely message.

It has been experimentally determined that the voting scheme works as well or better than a number of other measures that make use of the same information. For example, we tried the adjusted weight used by Teitelman \(^9\) in ARGUS, a hand printed character recognizer (a similarly structured pattern recognition system); and log \((1 + N_i)\) which is a good small sample probability estimate of the a posteriori probability of message \(i\).

**Functions and properties used in recognition**

The principal results of this research have been the development of properties which characterize speech input signals in a way to make recognition possible. Two different sets of recognition properties are included here. The first is a set of properties and functions which tend to describe the speech signal in more linguistic terms. Transitions of these properties describe features of the input message which can be understood in terms of the ordinary linguistic descriptions of such messages. In this way, they tend to more speaker independent than the second, more abstract set of properties which are described.

The second set of properties extract features of the spectral shape of a message. Their extreme simplicity makes them seem ideal for hardware implementation, and their high accuracy in recognition suggests that they capture most of the invariant qualities of our input message set.

**Linguistic features**

We distinguish *functions*, such as Loud\((i)\), which have a domain equal to the input spectral representation of an unknown word; *features*, such as Voice\((i)\), which are functions that are used in recognition and have a range limited to a small number of states; and *feature sequences*, such as Voice, which are the result of removing the time dimension from a feature. The following functions are used in the definitions of a set of 15 features.

**Functions**

1. Loud\((i)\) = \(F_1(i) + F_2(i) + F_3(i) + \text{MAX}[F_2(i), F_4(i)] + \text{MAX}[2(F_3(i) - F_4(i)), 0]\).

   This function is intended to provide a measure of the perceptual loudness of vowels. Perceptual loudness is related to vocal effort whereas the acoustic energy in a vowel depends in part on the vocal tract configuration as well as vocal effort. For example, the low vowel, \([a]\), for which the vocal tract is relatively open will have a greater natural intensity than the high vowel, \([i]\), for which the vocal tract is more closed (Lehiste and Peterson, 1959; Fant, 1960). The function is intended to compensate for the reduced level associated with a low first formant.

2. \(ah(i) = F_3(i) + F_4(i) - F_1(i) - F_2(i)\).

   This function and the following two functions are used to characterize the spectral quality of the vowel nuclei of a word. The vowels \([i, a, u]\) represent limiting articulatory positions for the tongue body. The vowel \([a]\), as in "pot," is produced by placing the tongue as far back and as low as possible without producing a constricted vocal tract (and therefore a consonant). An acoustic correlate of this articulation is a high first formant, resulting in a greater output in filters 3 and 4 than in filters 1 and 2. The function \(ah(i)\) will therefore be a maximum for vowels with a high first formant.

3. \(ee(i) = F_9(i) + F_{10}(i) + F_{11}(i) + F_{12}(i) - 2F_6(i) - 2F_7(i)\).

   The vowel \([i]\) as in "beet" is produced by positioning the tongue body as high and as forward in the mouth as possible. An acoustic correlate of this articulation is a low first and second formant, resulting in an energy minimum in filters 6 and 7, and an energy concentration between filters 9 and 12. The function \(ee(i)\) will therefore be a maximum for vowels similar to \([i]\).

4. \(oo(i) = F_3(i) + F_4(i) + F_5(i) - F_8(i) - F_{10}(i) - F_{12}(i) - F_{14}(i) - F_{16}(i)\).

   The vowel \([u]\) as in "boot" is produced by positioning the tongue body as high and as far back in the mouth as possible. An acoustic correlate of this articulation is a low first and second formant, resulting in a major energy concentration below \(F_7\) and reduced energy above \(F_7\). The function \(oo(i)\) will therefore be a maximum for vowels similar to \([u]\).

5. \(er(i) = F_7(i) + F_8(i) - F_4(i) - F_{12}(i)\).

   This function is similar to the vowel functions in form, but has the task of detecting spectra characteristic of the consonantal and syllabic allophones of \(r\). The low third formant of \(r\) and \(e\) (as the vowel in "bert") produces a distinct spectral shape with an energy concentration centered in filters 7 and 8, a dip in
energy at filter 4 between the first and second formants, and an absence of energy above F10 due to the low third formant. The function \( e_r(i) \) will therefore be a maximum when phones similar to [r, ə] are produced.

6. \( s_t(r(i)) = F_{16}(i) + F_{17}(i) + F_{18}(i) + F_{19}(i). \)

This function is a maximum when high frequency frication energy is present. The strident phones produce intense high frequency energy, whereas a non-strident fricative produces a small energy peak in filter 19. The function \( s_t(r(i)) \) will therefore be a maximum for many allophones of [s, z, ñ, ñ̃, ñ̃̃, j, t, k], which are the first consonants in [sin, zen, shin, azure, chin, gin, tin, kin] respectively.

7. \( d_{spe}t(i) = |ah(i) - ah(i - 1)| + |ee(i) - ee(i)| + |oo(i) - oo(i - 1)|. \)

This function is a computationally inexpensive approximation to a spectral derivative. The function will be a maximum when the spectrum is changing rapidly, as for example in a consonantal transition. \( d_{spe}t(i) \) is used as an aid in delimiting syllable boundaries. It is also used to distinguish ee-like vowels from some consonant-vowel transitions that produce a momentary peak in the function \( e_e(i) \).

8. \( c_v(i) = \text{sum of } F_1(i) \text{ through } F_{16}(i). \)

This function tends to be greater for vowels than for adjacent consonants. This is because a constricted or closed vocal tract configuration results in a reduced acoustic output in the frequency range spanned by these filters. This function is used to detect syllable boundaries.

9. \( d_{amp}(i) = \text{sum of } F_1(i) \text{ through } F_{19}(i) \text{ minus sum of } F_1(i - 1) \text{ through } F_{19}(i - 1). \)

This function indicates a sudden increase in energy in all filters. It will be a maximum for stop releases and sudden voicing onsets. It is used to detect stop bursts.

Features

The preceding functions are used in the computation of the following features. The usual interpretation of the three feature values (states) is: "−1" implies that the property is irrelevant for this time interval; "0" means that the property is relevant but not present; and "1" means that the property is relevant and present.

1. \( V_{o_i}(i) = 1 \text{ if } [F_2(i) > t_1] \) or \([F_2(i) > t_1 - 5] \) and \( V_{o_i}(i - 1) = 1 \)

0 otherwise

\( t_1 = 34 \)

A hysteresis region of 5 units is employed about the threshold, \( t_1 \), to ensure that a change of state is significant and not due to random fluctuations in level. The effect of the second conditional part of the definition is to keep \( V_{o_i}(i) \) in its current state for values of \( F_2(i) \) between 30 and 34, the hysteresis region.

An unknown word may vary considerably in overall recording level if the speaker moves with respect to the microphone or changes his vocal effort. It is much easier to compensate for recording level by modifying the threshold, \( t_1 \), than by attempting to modify the raw data. The maximum value attained by the time function \( L_{o_u_d}(i) \) is used for this normalization:

\[
M_{a_x_{l_{o_u_d}}} = \max[L_{o_u_d}(i)]
\]

\( L_{o_u_d}(i) \) attains its maximum value near the midpoint of the stressed vowel nucleus. For a speaker positioned correctly in front of a microphone and speaking at a comfortable level, a typical value for \( M_{a_x_{l_{o_u_d}}} \) is 200. Threshold normalization is accomplished by computing a new threshold, \( t_1' \), for each unknown word according to the following formulae:

\[
t_1' = t_1 * (200/M_{a_x_{l_{o_u_d}}})
\]

The method fails if the recording level increases to saturation or decreases below background noise. The effective range for recording level insensitivity is somewhat less than the full 45 decibel range of the input system.

The fundamental voice frequency of a male speaker will from about 90 to 180 Hz. The greatest energy due to voicing will appear in the low frequency filters because energy in the harmonics of the laryngeal source falls off at about 12 dB per octave. \( V_{o_i}(i) \) looks at the energy in filter 2. If this energy exceeds a threshold, voicing is present, otherwise not. This simple definition has worked surprisingly well. The greatest difficulty has been in treating final voiced stop releases consistently, and in detecting voicing energy in some voiced fricatives.

2. \( S_{y_{l_{a_{b}}}}(i) = 1 \) when the function \( e_v(i) \) is significantly increasing

0 when the function \( e_v(i) \) is significantly decreasing

otherwise, set to same value as in previous time sample.
The precise computation that implements this definition is too lengthy to be given here. The feature usually gives an accurate count of the number of syllables in a word and provides a rough segmentation. However, the times at which the feature changes state do not delimit precise syllable boundaries. Examples of difficulties with Syllable(i) include a frequently missed third syllable in "binary" and the segmentation into two syllables of some repetitions of the words "four" and "core."

3. Stress(i) =
   \[\begin{align*}
   & 1 \text{ if } \text{Loud}(i) = \text{Maxloud}\text{ for the first time,} \\
   & 0 \text{ if Syllable}(i) = 1 \\
   & -1 \text{ otherwise}
   \end{align*}\]
   
   Maxloud = Maxi [Loud(i)]

   This feature is designed to indicate which syllable of a word is the one which has the maximum loudness and is therefore the stressed syllable. Stress assignment has worked very well. The only word for which stress assignment has not consistently agreed with the expected stress assignment is "overflow," and this may be due to variations in the actual stress given the word by our speakers.

4. [a]-like(i) =
   \[\begin{align*}
   & -1 \text{ if } \text{Voice}(i) = 0 \\
   & 1 \text{ if } \text{[ah}(i) > 1 \text{ or } \text{[ah}(i) > -2 \text{ and } \\
   & [a]-\text{like } (i - 1) = 1 \\
   & 0 \text{ otherwise}
   \end{align*}\]

   This feature indicates the presence of the vowels [a, o, o′, , ae, A] in many phonetic environments (the vowels in [pot, boat, bought, bat, but] respectively).

5. [i]-like(i) =
   \[\begin{align*}
   & -1 \text{ if } \text{Voice}(i) = 0 \\
   & 1 \text{ if } \text{ee}(i) > 16 \text{ or } [\text{[i]-like } (i - 1) = 1 \\
   & \text{ and } \text{ee}(i) > 8 \text{ and } F19(i) < 25 \\
   & \text{ and } \text{Dspect}(i) < 30 \\
   & 0 \text{ otherwise}
   \end{align*}\]

   This feature indicates the presence of the vowels [i, I, o, e, o′, e] (the vowels in [beat, bit, bait, bet]) in many phonetic environments. The two additional clauses involving F19(i) and Dspect(i) are utilized to eliminate false responses to stridents and consonantal transitions which are momentarily [i]-like.

6. [u]-like(i) =
   \[\begin{align*}
   & -1 \text{ if } \text{[Voice}(i) = 0 \\
   & 1 \text{ if } \text{[oo}(i) > t2 \text{ or } [\text{[u]-like } (i - 1) = 1 \text{ and } \text{oo}(i) > t2-7] \\
   & \text{ and } F2(i) > F1(i) \\
   & 0 \text{ otherwise}
   \end{align*}\]

   \[t2 = 32\]

   This feature indicates the presence of the vowels [u, v, o, ] (the vowels in [boot, book, boat, bought]) in many phonetic environments. The additional clause involving F1(i) and F2(i) is intended to eliminate responses to consonants such as [m, n, l, w]. It has been found necessary to ignore any transitions to the "1" state in the features [i]-like(i) and [u]-like(i) if they last less than 6 time samples. Even with this additional constraint, a number of consonants are assigned one of these vowel-like qualities. In addition, there appear to be no natural threshold boundaries in the vowel-quality space so that a certain percentage of the vocabulary are treated inconsistently by each feature.

7. [r]-like(i) =
   \[\begin{align*}
   & -1 \text{ if } \text{[Voice}(i) = 0 \\
   & 1 \text{ if } [\text{[er}(i) > 18 \text{ and } \\
   & F3(i) - F4(i) > 5] \\
   & \text{ or } [\text{[er}(i) > 14 \\
   & \text{ and } [\text{r}-\text{like } (i-1) = 1]] \\
   & 0 \text{ otherwise}
   \end{align*}\]

   This feature indicates the presence of [r, ɛ] in many phonetic environments. The clause involving F3(i) and F4(i) is intended to ensure that the first formant frequency is not high. The low third formant of these phones produces a spectrum that is distinct and relatively easily differentiated from all other speech spectra. The feature works consistently except following voiceless stops and in some unstressed intervocalic positions.

8. Strident(i) =
   \[\begin{align*}
   & 1 \text{ when the function Str}(i) \text{ is significantly increasing} \\
   & 0 \text{ when the function Str}(i) \text{ is significantly decreasing} \\
   & -1 \text{ otherwise set to same value as in previous time sample}
   \end{align*}\]

   This feature indicates the presence of [s, z, s, z, t, d, k, g] in certain phonetic environments (generalol in stressed position and followed by a front vowel). A floating threshold is employed to divide a sequence of two successive stridents (e.g., [st]) into two strident segments separated by the state "—1."
9. Strid-stop(i) = \begin{cases} 
0 & \text{if } L\text{strident}(i) = 1 \\
1 & \text{if } Strident(i) = 1 \\
-1 & \text{otherwise}
\end{cases}

Where Lstrident(i) is the same as Strident(i) except that the state "+1" must occur for more than 6 time samples or it is transformed to the state -1. This property computes the length of a strident segment and places it in one of two categories. In general, a long strident is a continuant and a short strident is a stop. However, in some phonetic environments, voiceless stops may have an aspiration duration that is longer than the frication duration of an intervocalic fricative.

10. Fricative(i) = \begin{cases} 
1 & \text{if } [F19(i) > t4] \text{ or } [F19(i) > t4 - 4] \text{ and } [\text{Fricative}(i-1) = 1] \text{ and } [\text{Loud}(i) < 120] \\
0 & \text{if } \text{Voice}(i) = -0 \\
-1 & \text{otherwise}
\end{cases}

This feature looks for the high frequency energy characteristic of [f, v, θ, ð], (the initial consonants of [fin, vend, thin, then]). Strident phones generally exceed the threshold and are also detected by Fricative(i). The upper bound on Loud(i) is used to prevent a response to a loud vocalic sound in which energy is spread over the entire spectrum. A feature like Fricative(i) or Strident(i) sends a great deal of information about a word to the recognition algorithm, but neither feature is as consistent in its analysis of an unknown word as a simple feature such as Voice(i).

11. Stop-burst(i) = \begin{cases} 
1 & \text{if } [[\text{damp}(i) > 75] \text{ and } [F19(i-1) < 30] \text{ and } [F2(i) < 20] \text{ or } [F2(i) - F2(i-1) < 1]] \\
0 & \text{if } \text{Voice}(i) = 1 \\
-1 & \text{otherwise}
\end{cases}

This feature detects the sudden onset of energy in all filters that is characteristic of the release of a stop. The stop burst is distinguished from a rapid voicing onset by requiring the energy in F2(i) to either be low or not increasing. The "-1" state following a burst is removed if it is less than 3 time samples long. This transformation is an only partially successful attempt to signal voiced stops by the sequence "1, 0" and voiceless stops by the sequence "1, -1, 0."

12. Nasal(i) = \begin{cases} 
-1 & \text{if } \text{Voice}(i-2) = 0 \\
1 & \text{if } [\text{Loud}(i) - \text{Loud}(i-1) > 5] \text{ or } [\text{Nasal}(i-1) = 1] \\
& \text{and } \text{Loud}(i) - \text{Loud}(i-1) > 1] \\
& \text{and } ah(i) - ah(i-1) > 5 \\
& \text{or } ah(i) - ah(i-2) > 1] \\
0 & \text{otherwise}
\end{cases}

This feature looks for a voiced segment followed by a sudden increase in loudness and a rising first formant frequency. A nasal followed by [i] does not satisfy this criterion, but nasals in many other environments are detected. Initial [f] is frequently also detected by the feature. Features that look for dynamic properties of the input spectra are not easy to define efficiently. This relatively simple algorithm does not give a very satisfactory definition of nasality.

13–15. There are three other features of a somewhat different type. These features are intended to give a fairly precise characterization of the stressed vowel in the unknown word. The features, called [a]-stress, [i]-stress, and [u]-stress respectively, characterize the spectrum at a single time sample, that point during the stressed vowel when Loud(i) reaches a maximum for the word.

The ranges of the functions ah(i), ee(i), and oo(i) have been divided into 10 regions corresponding to the 10 possible states of the three special features:

<table>
<thead>
<tr>
<th>UPPER BOUND ON REGION</th>
</tr>
</thead>
<tbody>
<tr>
<td>State: 1 2 3 4 5 6 7 8 9 10</td>
</tr>
<tr>
<td>ah(i) none 14 9 4 0 -3 -6 -11 -16 -27</td>
</tr>
<tr>
<td>ee(i) none 80 60 40 20 0 -20 -40 -60 -80</td>
</tr>
<tr>
<td>oo(i) none 80 60 40 20 0 -20 -40 -60 -80</td>
</tr>
</tbody>
</table>

A typical stressed [i] phone might be characterized by the states [a]-stress = 9, [i]-stress = 1, and [u]-stress = 6. It has been found that this characterization is relatively stable for a given word. Most words fall into two or at most three states for each of the features.

Non-Linguistic Properties

A set of abstract properties was defined near the end of the research project in order to provide a comparative base for the performance of the linguistically motivated
features just described. These properties take advantage of some of the constraints that apply to speech spectra, but do not reflect the detailed phonetic content of the utterances.

There are two distinct types of spectral shape properties that were used in this set. The first nineteen properties crudely categorize the shape over time of the output of each of the nineteen filters, individually, with no attention paid to correlation between filters. There are three regions of "interest" in the output. The spectral property is given the value "-1" if the output is in the lowest region, "0" in the middle region, and "1" in the highest region. A hysteresis region around the transition threshold is used to prevent many "uninteresting" transitions when the filter output is near the border of these regions.

Formally, we define a set of 19 spectral amplitude properties, $S_n(i)$, as follows, where the notation $F_n(i)$ denotes the output of filter $n$ at the $i^{th}$ sample interval:

$$S_n(i) = \begin{cases} 1 & \text{if } [F_n(i) > B] \text{ or } [F_n(i) > B - 8] \text{ and } [S_{n(i)} - 1] = 1] \\ 0 & \text{if } [F_n(i) > A] \text{ or } [F_n(i) > A - 8] \text{ and } [S_{n(i)} - 1] \neq -1] \\ -1 & \text{otherwise} \end{cases}$$

A and B, as a function of the filter number, are given in the following table:

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>37</td>
<td>34</td>
<td>30</td>
<td>25</td>
<td>20</td>
<td>30</td>
<td>25</td>
<td>20</td>
<td>15</td>
<td>25</td>
<td>21</td>
<td>18</td>
<td>15</td>
<td>20</td>
<td>15</td>
<td>17</td>
<td>20</td>
<td>17</td>
<td>13</td>
</tr>
<tr>
<td>B</td>
<td>50</td>
<td>47</td>
<td>44</td>
<td>40</td>
<td>35</td>
<td>45</td>
<td>40</td>
<td>35</td>
<td>30</td>
<td>38</td>
<td>34</td>
<td>31</td>
<td>27</td>
<td>35</td>
<td>30</td>
<td>30</td>
<td>25</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

The threshold values in this table should be modified if there is a change in recording level. This can be done in the same way as previously described for thresholds involving linguistic features.

Another ten properties are used which roughly describe the correlation between adjacent filters for the lower ten filters. These properties categorize the differences between energies in adjacent filters. The three alternatives are: the next higher band has significantly more energy, significantly less energy, or about the same energy, as its lower neighbor. These properties are given values "1," "-1," and "0" respectively in these cases. Formally, we define a set of 10 spectral difference properties, $D_n(i)$, as follows, where $n = 1, 2, \ldots, 10$ and $m = n + 1$:

$$D_n(i) = \begin{cases} 1 & \text{if } [F_m(i) - F_n(i) > 4] \\ \text{or } [F_m(i) - F_n(i) > -1] \\ \text{and } [D_{n(i)} - 1] = 1]] \end{cases}$$

A change of state for one of these properties indicates that the spectral shape of the current time sample differs from the spectral shape of the previous time sample. A change of state is not produced by a simple change in level since it is the difference in energies which is significant. The large hysteresis region around the thresholds also prevents a property from changing state when there is no energy present in either filter.

The thresholds for the two types of properties were selected so that each state could be reached by at least some of the vocabulary items. The 29 properties produce sequences of states used in recognition. We can compare directly these recognition scores with those produced by state sequences from the 15 linguistic features.

**Results**

As benchmark experimental data, we recorded a word list, Table 2, used by Ben Gold in other speech recognition experiments. This list was recorded in a quiet room ($S/N$ ratio > 35 db) as spoken 10 times by two speakers, KS and CW. Both had been used as subjects by Gold. KS and CW were also recorded 5 times each on two other lists, an augmented International Word Spelling alphabet, Table 3, and a message list typical of one that might be used in a NASA mission context, shown earlier in Table 1.

The messages were recorded on high quality magnetic tape at 7½ ips, with approximately two seconds gap between words. For almost all experiments we used a digital tape containing the 6 bit logarithms of the filter bank output sampled every 10 milliseconds. Only in the noise degradation experiment was the analog input used directly. No difficulty was encountered in finding the beginning and end of each message.

A training round consists of one repetition of the entire word list in which the computer makes an identification attempt, is told the correct response, and stores the characterization of this message for use in future identification. With four rounds of training utilizing the linguistic properties, the system recognized correctly (gave as unambiguous first choice) 51 and 52 out of 54 words of the Ben Gold list for KS and CW respectively. Early experiments indicated that additional training would not increase the recognition scores after reaching its asymptote, and these 95% correct scores were available after only 3 rounds of training. Further experiments gave speaker KS 38 out of 38...
difficulty is that these later rounds were recorded at a different time. This indicates that some degradation in performance may be expected under less well controlled conditions.

Some incorrect responses are listed in Table 6 (for KS on the Gold list after 4 training rounds). These confusions give an indication of the types of phonetic information that is not well represented by the linguistic features. For example, the “four-core” confusion suggests that frication noise often falls below the lower thresholds of the filters, and that formant transitions are not sufficiently distinct to trigger different state transitions for this word pair. It is expected that some improvements could be made in the threshold settings by studying the data of Table 6.

<table>
<thead>
<tr>
<th>Previous Training</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Perfect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ben Gold List</td>
<td>43</td>
<td>44</td>
<td>49</td>
<td>51</td>
<td>47</td>
<td>45</td>
<td>49</td>
<td>51</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>Alphabet</td>
<td>30</td>
<td>36</td>
<td>34</td>
<td>38</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NASA List</td>
<td>35</td>
<td>37</td>
<td>43</td>
<td>43</td>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE 5—Typical recognition scores (KS)**

<table>
<thead>
<tr>
<th>Correct Answer</th>
<th>Guess</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weather at Splashdown Point</td>
<td>Alternate Splashdown Point</td>
</tr>
<tr>
<td>Four</td>
<td>Core, or Whole</td>
</tr>
<tr>
<td>Binary</td>
<td>Memory</td>
</tr>
<tr>
<td>Load</td>
<td>One</td>
</tr>
</tbody>
</table>

**TABLE 6—Typical mistakes**

These asymptotic recognition scores also indicate the separability of this 54 word vocabulary with these linguistic properties. The three vocabularies were combined into a single vocabulary of 114 distinct messages (duplications between vocabularies were considered only one message). On this larger vocabulary, the recognition score was a remarkable 110 out of 114 for KS. All those messages identified correctly in the individual lists were also identified in this larger context after the same four training rounds. This indicates very little interference between messages on these different lists; and implies that fairly large noninterfering (phonetically balanced) vocabularies might be constructed while maintaining this high recognition rate. It should be emphasized that while messages were repeated in training and testing, the same speech utterance was never seen more than once by the system.

To test the consistency of speakers uttering messages, we devised the spectral threshold properties described earlier. We tested these properties for recognition only
on the Ben Gold word list. After four rounds of training, the system was able to identify correctly 52 out of the 54 input messages presented to it for KS; and 51 out of 54 for CW. This rate is as good as that discussed for the linguistic features.

The spectral threshold properties appear to converge more rapidly to a higher recognition score on early training rounds. However, the linguistic properties may be slightly less speaker dependent. Neither of these tendencies is statistically significant so we are unable to make a definitive comparison.

There are 29 spectral threshold properties and only 15 linguistic properties. In addition, the spectral threshold properties tend to produce longer sequences. The comparable recognition rates of the two property sets suggest that the total information sent to the recognition algorithm by each set is roughly the same order of magnitude. If this is true, then it can be concluded that the linguistic sequences embody a more concise and more consistent representation of the vocabulary.

DISCUSSION

We have outlined the reasons why word recognition based on spectral input data is not a straightforward problem in pattern recognition. The invariants in the speech code appear to be relatively complex functions of the spectral input. We have also outlined the reasons why phoneme recognition is not an appropriate intermediate step in a word recognition algorithm. Unsolved problems include the segmentation of speech spectra into phoneme sized chunks and the derivation of rules for recognizing all of the allophones of a given phoneme. Our approach is intermediate between pattern recognition and phoneme recognition.

A set of acoustic feature detectors have been defined. The time dimension is removed from each feature so that the learning program deals only with the sequences of states produced by the feature definitions. The feature sequence characterization of an input message is sufficiently redundant so that the recognition algorithm is able to deal successfully with errors and inconsistencies in some fraction of the features.

The advantages of this approach are:

1. No precise segmentation of the utterance is required; features can change state in arbitrary time segments because the time dimension is removed from a feature in a way that is independent of the changes of state of other features.
2. A word need not be registered at exactly the same place in the 2 second speech sample; state sequences are independent of time origin.
3. Feature definitions can be made insensitive to the recording level by adjusting threshold settings according to a measure of the loudness of the unknown word.
4. Feature definitions can be made less sensitive to noise by employing hysteresis regions about thresholds.
5. State transition sequences weight changes in spectra as the important cues to word recognition (simple pattern-matching schemes will give undue weight to a sustained vowel).
6. Features may be defined to take advantage of natural acoustic boundaries between phones (Stevens, 1968b) and thereby minimize the variability in state transition sequences produced by a feature.
7. Features may be added to the system to provide redundancy for those decisions that are difficult. There is a direct control over the amount of redundancy sent to the recognition algorithm.
8. The feature approach permits the introduction and testing of linguistic hypotheses, such as placing greater emphasis on the properties of the stressed syllable.

The disadvantages of this approach are:

1. Removal of the time dimension discards all information concerning the simultaneous occurrence of specific states for two or more features. We have used the voicing feature to reintroduce some timing information. There exist pairs of English words for which this “time with respect to the voiced segments” is not enough to disambiguate the pair. Special features can always be defined to treat special cases, but our approach is not easily generalized to yield segmentation of an utterance into phone-sized chunks.
2. The features currently implemented are not speaker independent. Each speaker will have to train the system and this requires approximately 3 or 4 repetitions of the vocabulary.
3. Our system will degrade in performance as the length of the vocabulary is increased or as the number of speakers that it can simultaneously recognize is increased. This property is of course true of any recognition program; however, it should be noted that, with our current simple-minded set of features, there is a high error rate in any feature characterization and we rely heavily on redundancy to select the most likely input message.
4. For our limited objectives, the current implementation is computationally fast and gives satisfactory results. However, more sophisticated and more reliable features would be desirable.
The exponent in the function relating computational time and feature performance is not known but may be restrictively large.

A set of simple sum-and-difference properties was compared with the linguistically motivated features in the later stages of research. The comparable performance of this set of properties has led to several conclusions concerning the desirable attributes of features that operate within this framework.

A good feature includes a maximum amount of information in the sequences that it produces. Factors that influence information content are consistency of characterization and the number of vocabulary items that can be differentiated on the basis of the feature. A voicing feature that works perfectly does not contain as much information about the unknown word as any of the relatively inconsistent spectral properties. In other words, a moderate amount of inconsistency is tolerable if accompanied by increased word separability (and additional features to provide redundancy).

The spectral properties compare favorably on the basis of recognition scores because the linguistic features are fewer in number and computationally similar in form to the spectral properties. We believe that, even for limited vocabulary word recognition, a set of phonetically oriented features exist which are better in some sense than simple pattern features. The reasons for this faith consist of arguments that:

1. there exist natural acoustic boundaries between phones (Stevens, 20). Thresholds placed at these boundaries will produce features with more consistent sequence assignments. Features selected on this basis will be less sensitive to the free variations that occur in speech spectra.

2. there exist invariant attributes in acoustic waveforms from different speakers. These invariants, when incorporated into feature definitions, will produce recognition scores that are less sensitive to the individual speaker.

The arguments in favor of phonetically oriented features are offset by the computational simplicity of the spectral properties. Not enough is currently known in acoustic phonetics to take advantage of these theoretical benefits without additional basic research. We argue that a carefully selected set of properties like our spectral properties represent a practical word recognition solution that may not be superseded for some time to come.

The so-called spectral shape features were not selected primarily on the basis of their pattern recognition potential. They were carefully selected on the basis of what is known about the information bearing parameters of speech. An arbitrary set of parameters will not work (for the reasons outlined in the introduction). In that sense, the spectral shape features resemble acoustic phonetic parameters and are distinct from a number of other types of pattern characterizing functions that could have been chosen. Our results should not be interpreted to mean that a fundamental knowledge of speech is not needed when working on restricted problems such as limited vocabulary word recognition. On the contrary, it was only after developing a set of linguistic feature definitions within the context of our chosen recognition algorithm that we were able to devise a set of spectral shape functions possessing the necessary attributes.

ACKNOWLEDGMENTS

The authors wish to acknowledge the invaluable assistance of Lucille Darley who did most of the programming of LISPER, and Kenneth N. Stevens and Gottfried von Bismark, who designed and built the filter bank spectrum analyzer.

REFERENCES

1 D G BOBROW D L MURPHY W TEITELMAN
   The BBN LISP system
   BBN Report No 1677 April 1968
2 J MCCARTHY et al
   The LISP 1.5 programmers manual
   MIT Press 1964
3 J L FLANAGAN
   Speech analysis synthesis and perception
   New York Academic Press 1965
4 N CHOMSKY M HALLE
   The sound pattern of English
   New York Harper and Row 1968
5 R JAKOBSON G M FANT M HALLE
   Preliminaries to speech analysis
   The MIT Press Cambridge 1963
6 J HEMDAL G HUGHES
   A feature based computer recognition program for the modeling of vowel perception
   In W Wathen-Dunlop Models for the Perception of Speech and Visual Form pp 440-453 Cambridge The MIT Press 1964
7 B GOLD
   Word recognition computer program
   Technical Report 452 Research Laboratory of Electronics Cambridge MIT Press 1966
8 P DENES M MATHEWS
   Spoken digit recognition using time-frequency pattern matching
   The Journal of the Acoustical Society of America 32 pp 1450-1455 1960
9 H DUDLEY S BALASHEK
   Automatic recognition of phonetic patterns in speech
10 K H DAVIS H R BIDDOLPH S BALASHEK
   Automatic recognition of spoken digits
   The Journal of the Acoustical Society of America 24 pp 637-642 1952
11 G S SEBESTYEN  
Recognition of membership in classes  
IRE Transactions on Information Theory IT-6 pp 44–50 1961

12 B LINDBLOM  
Spectrographic study of vowel reduction  

13 L J GERSTMAN  
Classification of self-normalized vowels  
1967 Conference on Speech Communication and Processing  
6–8 November Office of Aerospace Research United States Air Force pp 97–100 1967

14 D B FRY P DENES  
The solution of some fundamental problems in mechanical speech recognition  
Languages and Speech 1 pp 35–58 1958

15 T B MARTIN A NELSON H ZADELL R COX  
Continuous speech by feature abstraction  
DDC No AFAL–78–66–189 Camden N J RCA 1966

16 D R REDDY  
An approach to computer speech recognition by direct analysis of the speech wave  
Technical Report C549 Palo Alto Cal Stanford University Computer Science Department 1966

17 G W HUGHES  
The recognition of speech by machine  

18 K N STEVENS  
Study of acoustic properties of speech sounds  

19 W TEITELMAN  
Real time recognition of hand printed characters  
Proc FJCC Spartan Press 1964

20 K N STEVENS  
The quantal nature of speech: evidence from articulatory-acoustic data  
In E E David and P B Denes eds Human communication A Unified View New York McGraw-Hill in press

21 K N STEVENS G von BISMARCK  
A nineteen-channel filter bank spectrum analyzer for a speech recognition system  