AD-779 306

AUTOMATIC VERIFICATION OF HYPOTHESIZED PHONEMIC STRINGS IN CONTINUOUS SPEECH

Richard A. Gillman

System Development Corporation

Prepared for:

Advanced Research Projects Agency

10 May 1974

DISTRIBUTED BY:

NTIS
National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TM-5315/000/00	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) Automatic Verification of Hypothesized Phonemic Strings in Continuous Speech		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(*s*) Richard A. Gillman		8. CONTRACT OR GRANT NUMBER(*s*) DAHC15-73-C-0080
9. PERFORMING ORGANIZATION NAME AND ADDRESS System Development Corporation 2500 Colorado Avenue Santa Monica, California 90406		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order #2254, Amdt. 1, Program Code 4D30
11. CONTROLLING OFFICE NAME AND ADDRESS Information Processing Techniques Office Advanced Research Projects Agency Arlington, Virginia		12. REPORT DATE 10 May 1974
		13. NUMBER OF PAGES 19
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)		15. SECURITY CLASS. (*of this report*) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Cleared for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 2215

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Speech understanding systems, phoneme recognition, acoustic processing

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

Procedures are described for mapping phonemic strings into a matrix of labeled speech segments that may consist of specific phonemes, groups of phonemes, or spectral classes. A computer program embodying these procedures is described.

TM-5315/000/00

AUTOMATIC VERIFICATION OF HYPOTHESIZED PHONEMIC STRINGS IN CONTINUOUS SPEECH

10 MAY 1974

RICHARD A. GILLMANN

THIS RESEARCH WAS SUPPORTED BY THE ADVANCED RESEARCH PROJECTS AGENCY OF THE DEPARTMENT OF DEFENSE UNDER CONTRACT NO. DAHC15-73-C-0080.

THE VIEWS AND CONCLUSIONS CONTAINED HEREIN ARE THOSE OF THE AUTHOR AND SHOULD NOT BE INTERPRETED AS NECESSARILY REPRESENTING THE OFFICIAL POLICIES, EITHER EXPRESSED OR IMPLIED, OF THE ADVANCED RESEARCH PROJECTS AGENCY OR THE U.S. GOVERNMENT

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVENUE • SANTA MONICA, CALIFORNIA 90406 • (213) 393-9411

II

Automatic Verification of Hypothesized Phonemic Strings in Continuous Speech

ABSTRACT

A procedure is described for mapping a phonemic string onto a computer representation of an interval of continuous speech. The procedure has as its inputs a hypothesized phonemic string and a matrix of labels and label scores obtained by computer analysis of a complete utterance. These two inputs are mapped by an iterative technique which (1) searches for individual phonemes in an appropriate order, not necessarily left-to-right; (2) continuously adjusts for phoneme duration variations; and (3) can deal with intraword coarticulation effects. The mapping procedure yields estimated time boundaries and a goodness-of-match score for the hypothesized phoneme string. A computer program has been written that uses this algorithm to perform automatic recognition of words and phrases within an operating continuous speech understanding system. Examples and current results from this system are presented.

TABLE OF CONTENTS

Section Page

1. INTRODUCTION

Phonemic-acoustic mapping is the process of matching a phonemic string to a computer analysis of a portion of a speech signal. This document describes the algorithms used in the mapping program of System Development Corporation's "Mini-System," a continuous-speech understanding system running entirely on a Raytheon 704 minicomputer. The Mini-System, which has been operational since May, 1973, was designed as a test platform for acoustic-phonetic routines now incorporated into an extended system that runs on an IBM 370/145.

2. OVERALL PROCESSING PATTERN OF THE MINI-SYSTEM

2.1 THE TOP-END

The Mini-System has a minimal linguistic top-end and powerful low-level acoustic analysis. The top-end is a simple left-to-right predictive parser with full backtracking. It begins by predicting all words that can begin an utterance, with the predicted left boundary for the list set to the beginning of the utterance. The rest of the system then returns to the top-end a subset of this list (possibly null), representing words judged to be possible acoustic matches, along with a goodness-of-match score and an estimated ending time boundary for each accepted word. The top-end then predicts the list of words possible syntactically after the best-matching first word, and so on. If at any point in this process all words in the predicted list fail to match, the top-end backtracks to a second-choice word found earlier.

2.2 THE A-MATRIX

The Mini-System operates as a two-stage process. The utterance is first analyzed by a set of acoustic-phonetic routines that produce a structured summary called the A-matrix. The second stage consists of the top-end, lexicon lookup, and mapping programs, which operate in a loop until the utterance is successfully parsed or all parsings fail. The Mini-System flow is illustrated in Figure 1.

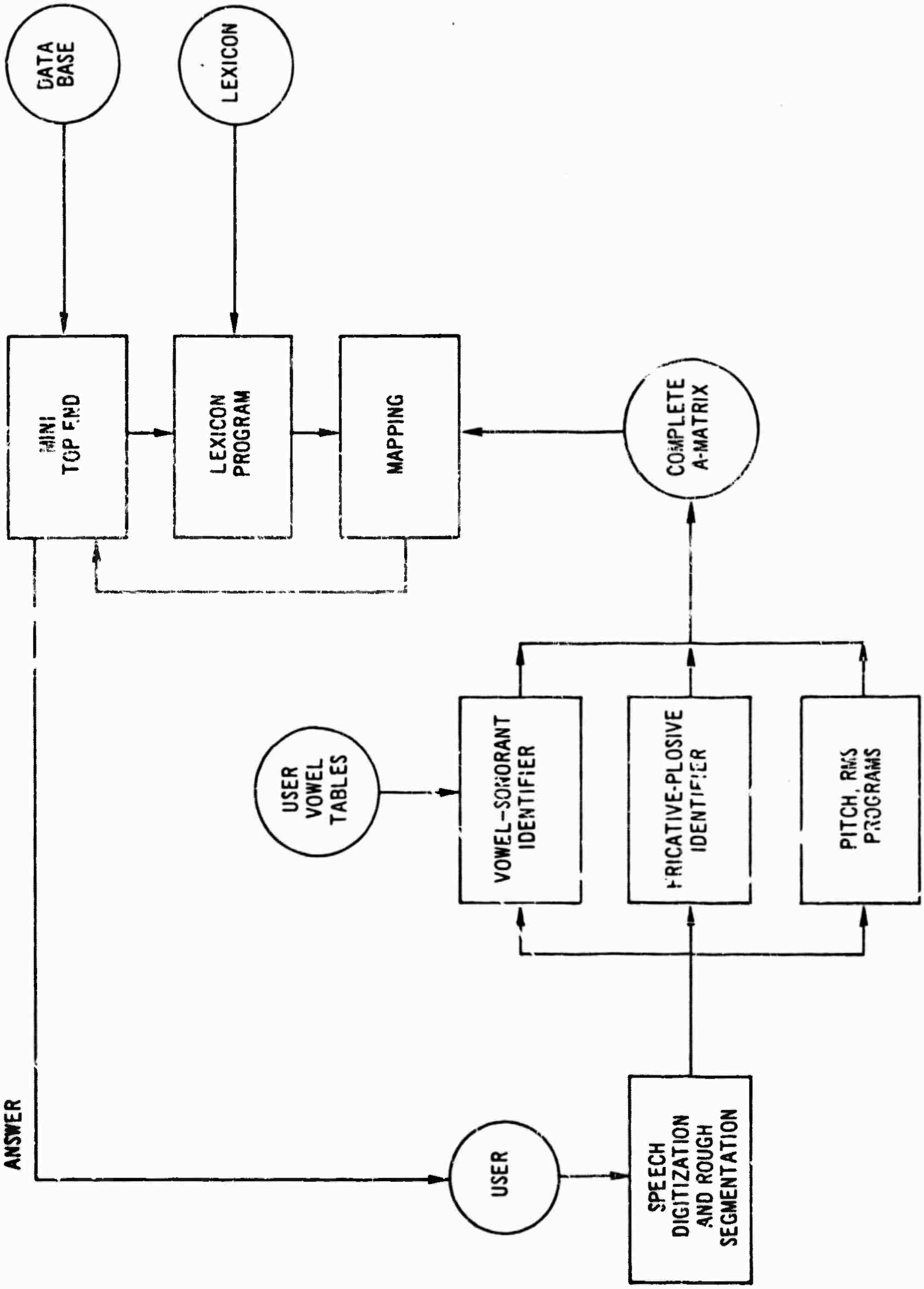

Figure 1. 704 Mini-System

The A-matrix is a two-dimensional array that contains the results of acoustic-phonetic analysis. The speech signal is divided into 10-msec. segments. For each segment, a set of labels and probabilities is created. The labels may be the names of specific phonemes, groups of phonemes, or spectral classes. In addition, pitch, energy, formant, and other data are recorded for each segment. (See Figure 2.)

2.3 THE MAPPING PROGRAM

The mapping program receives from the lexicon lookup program a list of words, orthographically spelled, with one or more phonemic spellings for each in a standard ARPABET notation (see Appendix 2.), and a beginning segment number. The mapping program then returns a subset, possibly null, of this list that represents words that are potentially present at the given point in the utterance. In addition, for those words judged possible, a word score, ranging from 0 to 99, is returned that represents the relative certainty of occurrence. The ending segment number for the word as estimated by the mapping program is also returned.

A problem arises when multiple phonemic variants of the same word are all judged to be possible occurrences. If all of the variants end at approximately the same place, the best is chosen and the others are ignored. If they end at different places, each must be kept as a separate possibility, because the top-end will have to consider each as a separate parse branch.

3. PHONEMIC-ACOUSTIC MAPPING

The following sections describe, in detail, the precise method used by the Mini-System mapping program to match phonemic strings against the A-matrix. The overall flow of the mapping process, illustrated in Figure 3, is to first deal with the word-juncture problem, thus obtaining an assumed beginning segment number for the word, then match each word in the predicted list to the A-matrix, deriving a goodness-of-match score and ending time boundary for each, and finally to subset the list of words on the basis of the match scores.

```
     EXPLA   PRINT
     SESSI.  NO.=    1       SAMPLE NO.=    3      SUBJECT NO.=14244
     SEG PS  1ST   2ND   3RD   4TH   5TH    RMS PITCH  F1   F2   F3  DB1  DB2  DB3  BW1  BW2  BW3  A1  Z1  A2  Z2  A3  Z3
       1 SI                                   1    0    0    0    0    0    0    0    0    0    0   0   0   0   0   0   0
       2 SI                                   1    0    0    0    0    0    0    0    0    0    0   0   0   1   0   0   0
       3 SI                                   1    0    0    0    0    0    0    0    0    0    0   1   0   1   0   0   0
       4 SI                                   1    0    0    0    0    0    0    0    0    0    0   0   0   1   0   0   0
       5 SI                                   1    0    0    0    0    0    0    0    0    0    0   0   0   1   0   0   0
       6 SI                                   0    0    0    0    0    0    0    0    0    0    0   0   0   1   0   0   0
       7 SI                                   0    0    0    0    0    0    0    0    0    0    0   1   0   1   0   0   0
       8 SI                                   1    0    0    0    0    0    0    0    0    0    0   1   0   1   0   0   0
       9 SI                                   0    0    0    0    0    0    0    0    0    0    0   1   0   1   0   0   0
      10 SI                                   1    0    0    0    0    0    0    0    0    0    0   0   0   1   0   0   0
      11 SI                                   1    0    0    0    0    0    0    0    0    0    0   0   0   1   0   0   0
      12 SI                                   1    0    0    0    0    0    0    0    0    0    0   1   0   1   0   0   0
      13 SI                                   0    0    0    0    0    0    0    0    0    0    0   1   0   1   0   0   0
      14 SI                                   0    0    0    0    0    0    0    0    0    0    0   0   0   1   0   0   0
      15 SI                                   1    0    0    0    0    0    0    0    0    0    0   0   0   1   0   0   0
      16 SI                                   0    0    0    0    0    0    0    0    0    0    0   0   0   1   0   0   0
      17 SI                                   0    0    0    0    0    0    0    0    0    0    0   1   0   1   0   0   0
      18 SI                                   1    0    0    0    0    0    0    0    0    0    0   1   0   1   0   0   0
      19 SI                                   0    0    0    0    0    0    0    0    0    0    0   0   0   1   0   0   0
    ┌ 20 VW AE 40 EH 25 AX 25    0           65  119  546 1679 2343  22   21   19  122  129  187  24   7  29  13  18  30
    │ 21 VW AE 40 AX 24 IH 18    0          163  108  546 1640 2255  25   23   18   89   83  187  48  10  90  34  55  53
    │ 22 VW AE 42 AX 19 IH 16    0          159  103  546 1679 2226  23   27   15   82   55  217  52  11  44  36  64  68
    │ 23 VW AE 48 AX 13 EH  9    0          119  111    0    0    0   0    0    0    0    0    0  58  11  38  33  36  53
    │ 24 VW AE 40 AX 13 EH  9    0          213  108    0    0    0   0    0    0    0    0    0  72  11  94  34  31  50
    │ 25 VW AE 48 AX 13 EH  9    0          230  107  565 1679 2148  29   24   13   58   86  436  69  13  54  33  55  60
 EH │ 26 VW AE 43 AX 13 EH  9    0          246  109    0    0    0   0    0    0    0    0    0  93  13  90  32  67  51
    │ 27 VW AE 48 AX 13 EH  5    0          327  108    0    0    0   0    0    0    0    0    0 100  13 151  34  93  54
    │ 28 VW AE 43 AX 13 EH  9    0          334  103    0    0    0   0    0    0    0    0    0  96  13 149  34  35  54
    │ 29 VW AE 45 AX 10 IH  9    0          330  135  546 1718 2109  27   29   21   71   93  216  75  13 161  35  87  55
    │ 30 VW AE 41 AX  7 IH  6    0          252  136  507 1716 2031  26   28   25   88   84  211  46   9 154  35  60  59
    │ 31 VW IH 37 AE  3 AX  3    0          105  101  429 1757 1553  22   30   29   95  105  133  19   9  73  37  23  47
    └ 32 VC IH 43 AE  2 VS  1 AX  1    0     47    0  390 1730 1914  21   32   33  149  137  133   3   3  36  37   6  25
    ┌ 33 UV VS  1                            14    0    0    0    0   0    0    0    0    0    0   3   4  11  31   2  10
    │ 34 UV PA 79 AP 19                       8    0    0    0    0   0    0    0    0    0    0   4   3   2  20   2   1
    │ 35 UV PA 79 AP 19                      17    0    0    0    0   0    0    0    0    0    0   2   6  10  31   5  25
 K  │ 36 UV PA 73 AP 19                      17    0    0    0    0   0    0    0    0    0    0   3   4  13  31   8  50
    └ 37 UV LB 56 PA 28 AP 14                19    0    0    0    0   0    0    0    0    0    0   2   4   7  32   7  52
    ┌ 38 SS AL 65 AP 33                      71    0    0    0    0   0    0    0    0    0    0   2   3  15  34  40  72
    │ 39 SS AL 55 AP 28 PA 14                102    0    0    0    0   0    0    0    0    0    0   1   1  10  34  42  74
    │ 40 SS AP 66 PA 33                     129    0    0    0    0   0    0    0    0    0    0   1   0  11  35  51  80
    │ 41 SS AL 66 AP 33                     109    0    0    0    0   0    0    0    0    0    0   1   1   5  34  41  72
 S  │ 42 SS AL 66 AP 33                     119    0    0    0    0   0    0    0    0    0    0   1   0   3  28  55  75
    │ 43 SS AL 56 AP 28 PA 14                102    0    0    0    0   0    0    0    0    0    0   1   0   4  32  35  75
    │ 44 SS AL 49 AP 24 PA 23                 71    0    0    0    0   0    0    0    0    0    0   1   0   3  28  24  58
    └ 45 UV PA 66 AP 33                      19    0    0    0    0   0    0    0    0    0    0   2   3   2   5  15  31
    ┌ 46 SI                                   2    0    0    0    0   0    0    0    0    0    0   0   0   1   0   2   2
    │ 47 SI                                   1    0    0    0    0   0    0    0    0    0    0   0   0   1   0   0   0
 P  │ 40 SI                                   1    0    0    0    0   0    0    0    0    0    0   1   0   1   0   0   0
    └ 49 SI                                   0    0    0    0    0   0    0    0    0    0    0   1   0   1   0   0   0
```

Figure 2. Sample Portion of an A-Matrix for the Beginning of the Word "EXPLAIN"

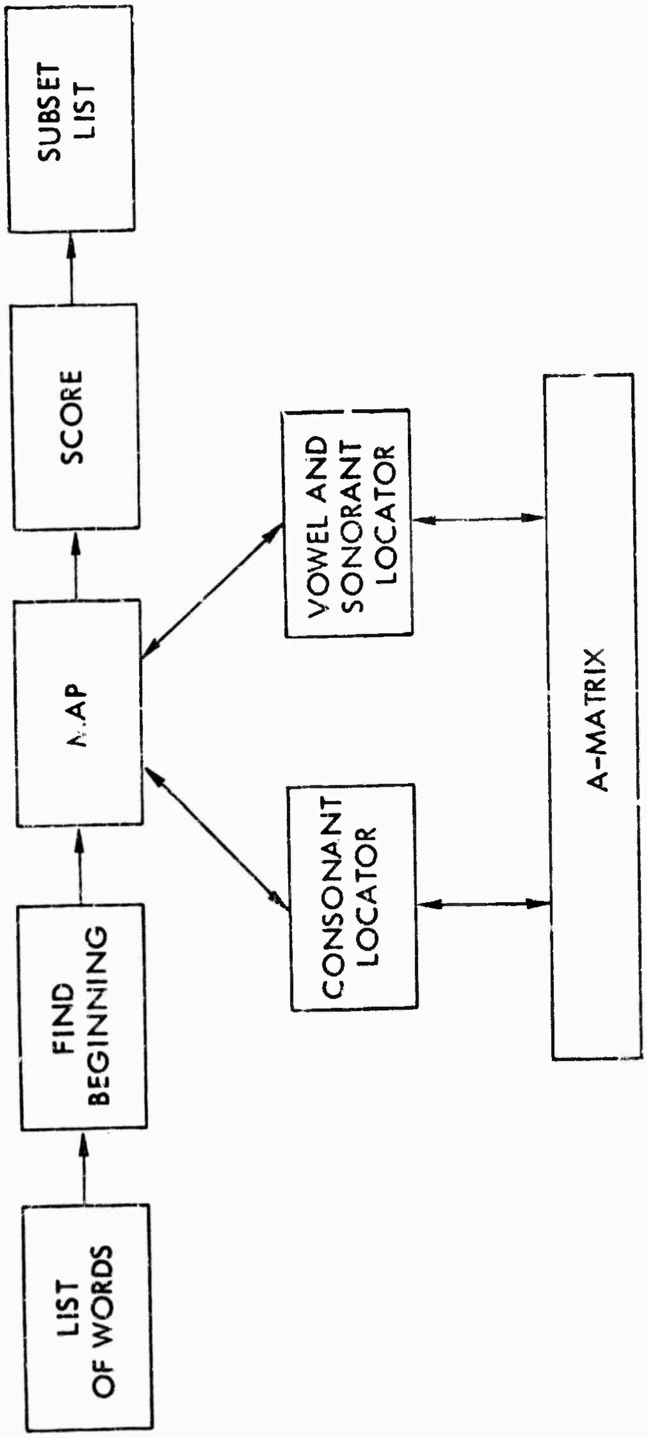

Figure 3. The Mapping Process

3.1 FINDING THE ASSUMED BEGINNING OF THE WORD

The mapping program receives from the top-end a segment number for the beginning of the word. This number represents the next segment after the end of the previous word found, and must be adjusted to take into account inter-word pauses, trailing aspirations, and other effects of word juncture. The algorithm used is based solely on rough segment labels, which are approximate spectral labels assigned every segment based on amplitude and zero-crossing data. Five rough segment labels are possible: SI (silence), UV (low-amplitude voiced or unvoiced), VW (vowel-like), SS (strong frication) and VC (all other--usually weak voicing). The following algorithm is used to adjust the assumed beginning.

B1. Ignore Trailing Aspiration

If the next four segments are SI, UV, or VC segments, and are followed by at least six SI or UV segments, advance the beginning seven segments.

B2. Skip Over Long Pauses

Advance the beginning over any SI segments, ignoring isolated UV, SS, or VC segments.

B3. Ignore Tongue Clicks

If the next five segments are followed by at least eleven consecutive SI segments, advance the beginning fourteen segments and go to step B2.

B4. Back Off

Back off two segments unless that would retreat the assumed beginning to before the beginning given by the top-end. This is to ensure that plosive bursts and such have not been inadvertently skipped over.

3.2 THE MAPPING TABLE

A special table is used throughout the mapping process to keep track of intermediate results. This is a sample mapping table for the word "PRINT":

PHX	STX	CLX	SCX	FNX	BX	EX
P	0	PL	20	1	10	14
R	0	SO	0	0	15	17
IH	2	VW	34	1	18	30
N	0	NA	0	0	31	38
T	0	PL	0	0	39	46

The columns have the following meanings:

PHX The name of the phoneme in standard two-character ARPABET.

STX The predicted stress. Zero means reduced, one means unstressed, and two means stressed. Stress is meaningful only for vowels.

CLX The phoneme class. The following classes are used:

VW: IY, IH, EY, EH, AE, AA, AH, AO, OW, UH, UW, AX, IX, ER, AW, AY, OY, EL, EM, EN
NA: M, N, NX
SO: Y, W, R, L
SS: S, SH, Z
PL: P, T, K, B, D, G
XX: All others

SCX The score, or probability of occurrence, of the phoneme as judged by the phoneme locators, has a range of 0 to 99. SCX is meaningful only if FNX=1.

FNX If FNX is zero, the phoneme locators have not yet been called for this phoneme. If FNX is one, SCX contains the phoneme score determined by the phoneme locators.

BX
EX These represent the beginning and ending 10-msec segment numbers for each phoneme. If FNX=1 and SCX>0, BX and EX are as found by the phoneme locators. Otherwise, they are estimated positions set by the mapping program.

Having located the assumed beginning of the word, an initial table is created from the ARPABET spelling received from the lexicon lookup program. The PHX column is filled with phoneme names, the STX column is filled with stress data, and the CLX column is filled with the appropriate phoneme classes. SCX and FNX are set to zero. The beginning and ending segments for each phoneme, BX and FX, are filled, starting from the assumed beginning and allotting 8 segments (80 msec.) to each phoneme. The following is a sample initial mapping table for the word "PRINT," using an assumed beginning segment number of 10:

PHX	STX	CLX	SCX	FNX	BX	EX
P	0	PL	0	0	10	17
R	0	SO	0	0	18	25
IH	2	VW	0	0	26	33
N	0	NA	0	0	34	41
T	0	PL	0	0	42	49

3.3 THE PHONEME LOCATORS

The mapping process uses a set of subroutines that locate a given phoneme near a given estimated position in the speech utterance and return a probability-of-occurrence estimate for that phoneme and location; these subroutines employ the acoustic results stored in the A-matrix. The phoneme locators perform their search in a two-step fashion: first, they look for the phoneme strictly within or overlapping the interval given them by the mapper. If this fails, they then expand their search by a certain "fuzz" distance, which is currently set to three segments. At the present time, two phoneme locators are used, one for vowels and sonorants and one for consonants.

The vowel and sonorant locator looks for the vowels and the phonemes Y, W, R, L, M, N, and NX. When looking for a diphthong, the locator looks for an appropriate pattern of change in the A-matrix. For the other phonemes, the decision is based on the choices recorded in the A-matrix and on considerations of acoustic similarity. This locator is also capable of considering a certain amount of coarticulation effects by using information on neighboring phonemes as recorded in the mapping table. If an interval is given that is voiced but significantly unlike the requested phoneme, a score of one is returned.

The consonant locator looks for all the phonemes not covered by the vowel and sonorant locator. The high-energy fricatives S, SH, Z, CH, JH, and ZH are located in a straightforward manner. The stop consonants P, T, B, K, G, and D are identified in the A-matrix only at the plosive burst; thus, the locator must allocate a certain portion of any preceeding silence to them. A low-energy area not identified as a particular low-energy phoneme will get a nominal score when a low-energy phoneme is requested.

3.4 ORDERING OF THE PHONEME SEARCH

A "goodness score" is associated with each phoneme in the mapping table. This number is used to decide which phoneme to look for next. Each goodness score is proportional to distance from the nearest fixed boundary, in phonemes, and the difficulty of locating the phoneme by machine in the acoustic signal. The function below is currently used to create these scores. Naturally, as our ability to recognize certain phonemes improves, this function is changed to reflect that fact.

Goodness =	(Distance + 1)	If the phoneme is a stressed vowel, or a member of the SS class, or an N, or an R, or an ER.
	(Distance +1)*2	If the phoneme is an unstressed or reduced vowel, or a Y, or a W, or an M, or an NX.
	(Distance +1)*3	If the phoneme is a member of the PL class, or a CH, or a JH, or an L.
	(Distance +1)*5	otherwise.

These scores are used to order the phoneme search. The phoneme with the lowest score is looked for first, then the one with the second-lowest score is looked for, and so on. In this way we avoid the problems of a strictly left-to-right search, which may stumble over a difficult phoneme, while on the other hand not plunging so far into the word, looking for an easy phoneme, that we give up a decent estimate of the phoneme's location.

3.5 THE MAPPING ALGORITHM

After the assumed beginning is calculated and an initial mapping table is created, the actual phonemic-acoustic mapping process is performed.

First, an anchor phoneme is selected, as follows. The goodness score for each phoneme is calculated, and the two lowest-scoring phonemes are given to the phoneme locators, along with the beginning and ending estimated segment boundaries recorded in the mapping table. The phoneme returning the higher score is selected as the anchor.

The anchor phoneme is entered into the mapping table as found. The estimated phoneme locations of all the phonemes in the table are adjusted to reflect the new information on the boundaries of the anchor phoneme. Mapping now proceeds in an analogous manner, with the unlooked-for phoneme having the best goodness score being looked for next. After each phoneme is looked for, the table is adjusted to reflect new boundary data, and the goodness scores are recomputed. This process continues until all phonemes in the word have been looked for.

During the mapping, as each phoneme is located, the estimated boundaries for phonemes not yet located must be adjusted to reflect any new information. This is done as follows. Assume we have a partially filled mapping table with some phonemes already found with definite boundaries, some looked for but not found and some not yet looked for. We are now given the result of a call to a phoneme locator, i.e. a score, a beginning segment number, and an ending segment number. If the new answer has a score of zero, meaning that the phoneme was not found, the boundaries estimated previously are not changed. If the new answer is positive, a check is made to ensure that the new boundaries are consistent with previous results. If the new phoneme occurs before the last previously found phoneme or after the next previously found phoneme, the new answer is rejected; otherwise, the new beginning and ending segment data are entered into the mapping table. Note that this allows considerable overlap to occur between phonemes but that they are not allowed to be out of order in time.

At this point, boundaries of the unlocated phonemes must be adjusted. First, the boundaries of phonemes that lie between two known boundaries are adjusted to agree with the fixed data. If two or more phonemes lie between the same fixed boundaries, each is given an equal portion of the available slot. Finally, the unlocated phonemes occurring after the last fixed time boundary and before the first fixed time boundary are shifted forward or backward to agree with that boundary, with 8 segments (80 msec.) allotted to each.

At the end of the mapping process, after all phonemes have been looked for, a second pass is performed. That is, if any phonemes were not found (score of zero), they are looked for again, on the supposition that estimates of their boundaries may have improved during the mapping of other phonemes. This second pass is repeated until no further progress is made.

At this point, the mapping table is complete and ready to be scored.

3.6 SCORING THE COMPLETED MAPPING TABLE

An overall word score is derived from the completed mapping table. Originally, this was just the average phoneme score, but further development has complicated the scoring algorithm. This is the scoring algorithm currently in use:

S1 SUM, DVSR = 0. I = 1.

S2 If the word begins more than 8 segments past the assumed beginning, DVSR is incremented by 1 for each such 8 segments.

S3 Any more phonemes left to score? If not, go to step S9.

S4 Add SCORE(PHONEME(I)) to SUM. Add 1 to DVSR.

S5 If PHONEME(I) is a stressed vowel, repeat step S4.

S6 If SCORE(PHONEME(I))=0, add 1 to DVSR for each 4 segments in excess of 8 segments of length.

S7 If 0<SCORE(PHONEME(I))<5, add 1 to DVSR for each 4 segments in excess of 12 segments of length.

S8 I=I+1. Go to step S3.

S9 If LENGTH(WORD) is not at least 3 segments times the number of phonemes in the word, add 1 to DVSR for each segment under this minimum.

S10 SUM=2*SUM. DVSR=MAX(DVSR,1).

S11 SCORE=MIN(SUM/DVSR,99).

S12 Multiply the score by the fraction of phonemes in the word having a positive score.

4. CONCLUSIONS

The approach to phonemic-acoustic mapping described in this document has five important advantages: (1) phonemes are searched for in an order that maximizes the chance that they will be properly found; (2) the process is able to deal with phoneme duration variations without resorting to arbitrary thresholds for minimum and maximum phoneme length; (3) by performing segmentation in conjunction with recognition, a more accurate segmentation is possible than by segmentation in isolation; (4) the procedure is capable of correctly identifying words even when it cannot identify every single phoneme in the word--i.e., it is fail-soft; and, (5) it is extensible: to add a new word to the vocabulary, we add only its phonemic spelling, not any code.

In a trial run of 20 utterances, the program was able to identify the correct word in first place 89% of the time. The majority of errors resulted from co-articulation effects between words, which are beyond the scope of the top-end of the Mini-System. The vocabulary for this test was 160 words, and the grammar contained 30 rewrite rules.

APPENDIX 1: EXAMPLE MAPPING OF THE WORD "EXPLAIN"

a) Initial mapping table

PHX	STX	CLX	SCX	FNX	BX	EX	(Goodness)
AX	0	VW	0	0	15	22	4 **
K	0	PL	0	0	23	30	9
S	0	SS	0	0	31	38	4 **
P	0	PL	0	0	39	46	15
L	0	SO	0	0	47	54	18
EY	2	VW	0	0	55	62	7
N	0	NA	0	0	63	70	8

Score of "AX" is 29

Score of "S" is 60 -- therefore "S" chosen as anchor.

b) Mapping table after the anchor "S" is entered.

PHX	STX	CLX	SCX	FNX	BX	EX	(Goodness)
AX	0	VW	0	0	16	23	4 **
K	0	PL	0	0	24	31	6
S	0	SS	60	1	32	44	x
P	0	PL	0	0	45	52	6
L	0	SO	0	0	53	60	9
EY	2	VW	0	0	61	68	4
N	0	NA	0	0	69	76	5

c) The "AX" is now searched for and entered.

PHX	STX	CLX	SCX	FNX	BX	EX	(Goodness)
AX	0	VW	29	1	17	22	x
K	0	PL	0	0	23	31	6
S	0	SS	60	1	32	44	x
P	0	PL	0	0	45	52	6
L	0	SO	0	0	53	60	9
EY	2	VW	0	0	61	68	4 **
N	0	NA	0	0	69	76	5

d) The program now leaps out for the "EY" and finds it.

PHX	STX	CLX	SCX	FNX	BX	EX	(Goodness)
AX	0	VW	29	1	17	22	x
K	0	PL	0	0	23	31	6
S	0	SS	60	1	32	44	x
P	0	PL	0	0	45	54	6
L	0	SO	0	0	55	63	6
EY	2	VW	35	1	64	81	x
N	0	NA	0	0	82	89	2 **

e) The program now looks for and finds "N".

PHX	STX	CLX	SCX	FNX	BX	EX	(Goodness)
AX	0	VW	29	1	17	22	x
K	0	PL	0	0	23	31	6 **
S	0	SS	60	1	32	44	x
P	0	PL	0	0	45	54	6
L	0	SO	0	0	55	63	6
EY	2	VW	35	1	64	81	x
N	0	NA	34	1	82	93	x

f) The program now returns to fill in the "K".

PHX	STX	CLX	SCX	FNX	BX	EX	(Goodness)
AX	0	VW	29	1	17	22	x
K	0	PL	30	1	24	31	x
S	0	SS	60	1	32	44	x
P	0	PL	0	0	45	54	6 **
L	0	SO	0	0	55	63	6
EY	2	VW	35	1	64	81	x
N	0	NA	34	1	82	93	x

g) The mapper now proceeds to the plosive "P".

PHX	STX	CLX	SCX	FNX	BX	EX	(Goodness)
AX	0	VW	29	1	17	22	x
K	0	PL	30	1	24	31	x
S	0	SS	60	1	32	44	x
P	0	PL	19	1	45	53	x
L	0	SO	0	0	54		6 **
EY	2	VW	35	1	64	81	x
N	0	NA	34	1	82	93	x

h) The mapper now looks for the "L" and fails to find it. Thus the final mapping table is:

PHX	STX	CLX	SCX	FNX	BX	EX
AX	0	VW	29	1	17	22
K	0	PL	30	1	24	31
S	0	SS	60	1	32	44
P	0	PL	19	1	45	53
L	0	SO	0	1	54	63
EY	2	VW	35	1	64	81
N	0	NA	34	1	82	93

This evaluates to a final word score of 51.9.

APPENDIX 2: THE ARPABET COMPUTER PHONEMIC ALPHABET

COMPUTER PHONETIC REPRESENTATIONS 8/1/73

Phoneme	Computer Representation		Example	Phoneme	Computer Representation		Example
	1-Character	2-Characters			1-Character	2-Characters	
i	i	IY	beat	p	p	P	pet
I	I	IH	bit	t	t	T	ten
e	e	EY	bait	k	k	K	kit
ɛ	E	EH	bet	b	b	B	bet
æ	@	AE	bat	d	d	D	debt
ɑ	a	AA	Bob	g	g	G	get
ʌ	A	AH	but	h	h	HH	hat
ɔ	o	AO	bought	f	f	F	fat
o	o	OW	boat	θ	T	TH	thing
U	U	UH	book	s	s	S	sat
u	u	UW	boot	š or ʃ	S	SH	shut
ə	x	AX	about	v	v	V	vat
ɨ	X	IX	roses	ð	D	DH	that
ɝ	R	ER	bird	[illegible]	z	Z	zoo
aU or aw	W	AW	down	ž or ʒ	Z	ZH	azure
aI or ay	Y	AY	buy	č	C	CH	church
ɔI or ɔy	O	OY	boy	ǰ	j	JH	judge
y	y	Y	you	ʍ	H	WH	which
w	w	W	wit	syl l,l̩	L	EL	battle
r	r	R	rent	syl m,m̩	M	EM	bottom
l	l	L	let	syl n,n̩	N	EN	button
m	m	M	met	flapped t,ɾ	F	DX	batter
n	n	N	net	glottal stop,ʔ	Q	Q	
ŋ	G	NX	sing	silence	-	-	
				non-speech segment	!	!	laugh, etc.

AUXILIARY SYMBOLS (1- AND 2-CHARACTER CODES ARE IDENTICAL)

Symbol	Meaning	Symbol	Meaning
+	Morpheme boundary	:3 or .	Fall-rise or non-term juncture
/	Word boundary	* **	Comment (anything except * or **)
#	Utterance boundary	' '	Apos-surround special symbol in comment
:	Tone group boundary		
:1 or .	Falling or decl. juncture	( )	Phoneme class information
:2 or ?	Rising or inter. juncture	< >	Phonetic or allophonic escape

STRESS REPRESENTATIONS (IF PRESENT, MUST IMMEDIATELY FOLLOW THE VOWEL)

Value	Stress Assignment	Value	Stress Assignment
0	No stress	3	Tertiary stress
1	Primary stress	.	(Etc.)
2	Secondary stress	. :	

NOTE: Spaces are ignored except within escapes.