# Automatic Text Alignment for Speech System Evaluation

JOSEPH PICONE, MEMBER, IEEE, KATHLEEN M. GOUDIE-MARSHALL, GEORGE R. DODDINGTON, MEMBER, IEEE, AND WILLIAM FISHER, MEMBER, IEEE

*Abstract*—**This paper describes an algorithm which automatically aligns two text strings, maximizing the phoneme-to-phoneme similarities between the strings. The alignment problem is formulated in an unconstrained endpoint dynamic optimization framework, and operates on the phonemic transcriptions of the input text strings. A modification of the standard dynamic programming approach allows insertion and rejection errors in either string to be correctly identified. Benchmark evaluations show the performance of the algorithm approaches "the expert speech scientist." The algorithm is an ideal tool for automated evaluation of experimental data, including data collected from speech intelligibility experiments and speech recognition experiments.**

## I. Introduction

SPEECH intelligibility testing typically requires a listener to transcribe the perceived contents of some speech stimuli [1], [2]. From this transcription, an error rate can be computed in terms of insertion, rejection, and substitution errors at either the word or phoneme level. Similarly, evaluating the accuracy of a connected word speech recognizer requires performing the same type of computation on the output of a speech recognizer. In either case, it is very important to have an accurate and repeatable means of scoring such experiments.

The algorithm presented in this paper automatically aligns two text strings, one defined as the stimulus string, representing the data presented for testing, and the second defined as the response string, representing the output from some experiment which corresponds to the stimulus string. The alignment is performed on the phonemic transcriptions of these text strings. From the alignment, a phonemic error rate is computed in terms of substitution, rejection, and insertion errors. An important intermediate output of this alignment is a phoneme confusion matrix which displays the actual confusability of each phoneme. This information has proven useful in the development of speech processing systems, since it can be used to interpret the nature of the errors which occur in the evaluation of a system [3], [4]. The phoneme-level alignment is translated into a word-level alignment, which is also scored for substitutions, rejections, and insertions of words.

Manuscript received May 6, 1985; revised September 24, 1985.

The authors are with Texas Instruments, Inc., Speech Systems Research, Dallas, TX 75266.

IEEE Log Number 8608126.

The basic problem associated with text alignment is the definition of a meaningful distance metric between two text units, such as words or phonemes, such that the degree of similarity between the two strings can be maximized. Any similarity measure used in an automated scoring algorithm must be a perceptually based measure. It is important that the output of the algorithm accurately interpret the listeners' impressions of the stimulus data. For instance, two homophones differ in spelling, yet are identical phonemically (for example, "scent" and "cent"). Puns play on the similarity of sounds in words while having radically different spellings and meanings ("to wreck a nice beach" and "to recognize speech"). It is not clear how text strings can be aligned using only the raw text. Our approach is to accommodate these problems by performing matching at the phoneme level. Phoneme-to-phoneme distances can then be computed in a perceptually meaningful way based on experimentally derived phoneme-to-phoneme distances which have been collected through various listening experiments [5], [6].

Another major problem in developing an automated algorithm is inaccurate computation of insertion and rejection errors. Any alignment algorithm must be able to reject an arbitrary number of words in either string. This is a difficult problem to deal with in a typical dynamic programming approach, in which discontinuous warping paths are not allowed. We have developed a modification of the standard unconstrained endpoint dynamic optimization problem which allows insertion and rejection errors at any stage of the optimization.

Only two constraints are imposed on the matching procedure. First, the order of each input text string is preserved. Errors such as transposition errors can only be accounted for through postprocessing techniques. Second, each phoneme in the stimulus string can match to only one phoneme in the response string. Many-to-one mappings are not allowed, although in some cases such a mapping may be desirable.

The combination of this modified dynamic programming algorithm and the phonemic-based matching has resulted in an algorithm which approaches the performance of the expert speech scientist. The details of the alignment algorithm are presented in Section II. In Section III, the output of the automated alignment algorithm is compared to data supplied by three "expert speech scientists." Fi-

0096-3518/86/0800-0780$01.00 

nally, in Section IV, we conclude this paper with a discussion of several possible refinements of the algorithm.

## II. Phonemic-Based Text Alignment

The kernel of this algorithm is a string alignment procedure which operates on a phonemic transcription of the input text strings. The incoming strings are converted to phonemes using a text-to-phoneme system developed at Texas Instruments [7]. The phoneme set is defined in Table I. The particular set of production rules [7] used in this translator achieves an accuracy of 99.1 percent correct phonemes, measured using the 10 000 most frequent English words as a test database, weighted with word frequencies.

The alignment which is produced by this algorithm relies heavily upon the accuracy and consistency of the text-to-phoneme translator. Since the translator produces a very broad transcription which does not reflect particular idiosyncrasies of the underlying speech data, such as coarticulation effects or varying pronunciations of a given word, it is possible to obtain alignments which at best only approximate the underlying perceptual error. While provisions have been made for a human linguist to manually correct errors in the text to phoneme translation, for run of the mill testing, the present level of performance of the text to phoneme translator in [7] appears to be sufficient.

After both input strings are transcribed, the response phoneme string is mapped onto the stimulus string using a modified unconstrained endpoint dynamic optimization algorithm. A time constraint is imposed such that the time order of the phonemes of each string must be preserved. A typical dynamic optimization scenario is shown in Fig. 1. Here, the stimulus string is "a test," while the response string is "the best test."

The transcriptions are shown in Fig. 1, displayed in a grid form, similar in format to a dynamic time warping formulation. Our convention is to let the horizontal direction denote the response string, and the vertical direction denote the stimulus string. Implicit in this formulation is the notion of a "null" phoneme, that is, a phoneme which will denote an insertion or rejection error. Note that the last phoneme in the phoneme set defined in Table I contains a symbol "**" which denotes the null phoneme. Also, observe that each phoneme string in Fig. 1 has been prepended and appended with the null phoneme. These two additional nodes are used as anchors in the sense that the final optimal path determining the alignment of the two strings will always pass through these two points.

At each stage in the optimization, a cost function is computed for all previous nodes including the root node (0, 0). The dynamic programming cost function is defined as

$$C(i, j) = D(i, j) + \min_{m, n}\left\{C(m, n) + \sum_{p=m+1}^{i-1} D(p, 0)\right\}, \quad (1)$$

TABLE I
A List of Phoneme Symbols

| Phoneme | Example | Phoneme | Example |
|---|---|---|---|
| IY | beat | F | fat |
| UW | boot | V | vat |
| EY | bait | M | met |
| OW | boat | T | ten |
| AE | bat | D | debt |
| AA | bob | DX | batter |
| AY | buy | TH | thing |
| AW | down | DH | that |
| OY | boy | N | net |
| AO | bought | S | sat |
| IH | bit | Z | zoo |
| UH | book | CH | church |
| EH | bet | JH | judge |
| AX | about | SH | shut |
| ER | bird | ZH | azure |
| Y | you | K | kit |
| W | wit | G | get |
| R | rent | NX | sing |
| L | let | HH | hat |
| P | pet | WH | which |
| B | bet | ** | null word |

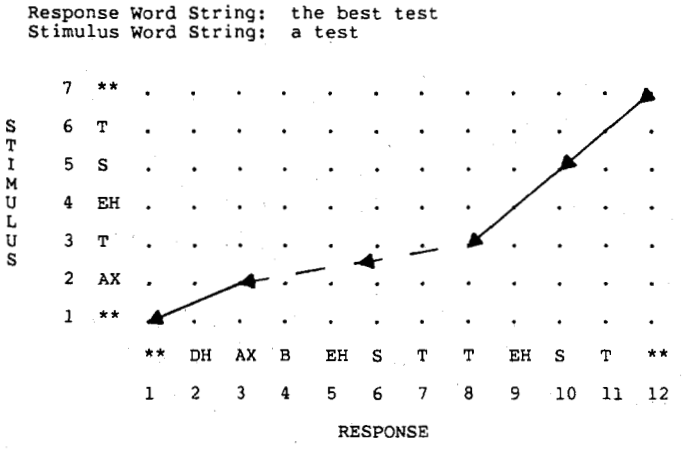



Fig. 1. An alignment example showing insertion errors. (The dashed line indicates insertion errors.)

where $D(i, j)$ denotes the distance between the $i$th and $j$th phoneme, and $C(m, n)$ denotes the accumulated error at node $(m, n)$. $D(p, 0)$ denotes the distance from the $p$th phoneme to the null phoneme. The cost function of (1) differs from the standard dynamic programming formulation in that the cost at node $(i, j)$ is actually a minimum over the region defined by $m = 0$ and $n = 0$, and $1 \leq m \leq i - 1$ and $1 \leq n \leq j - 1$. The values of $m$ and $n$ which produce the minimum cost are chosen as the back pointer at node $(i, j)$. Note that if $m \neq i - 1$ or $n \neq j -$

1, an insertion or deletion error will occur (if this back pointer is chosen as a part of the optimal path).

The cost function of (1) is computed for all nodes in the graph of Fig. 1, beginning with node (0, 0), and concluding with the terminal node ($ns + 1$, $nr + 1$), where $ns$ denotes the number of phonemes in the stimulus string, and $nr$ denotes the number of phonemes in the response string. The optimal path represents the path which minimizes $C(ns + 1, nr + 1)$.

Observe that this formulation differs from the standard dynamic programming in that a back pointer consists of two indexes, and all previous hypotheses are searched at any node. An insertion error occurs when any phoneme in the response string is left unused (that is, the optimal path skips a column). This unused phoneme is matched to the null phoneme for diagnostic purposes. Rejection errors occur when any phoneme in the stimulus string is left unused. The second term in (1) represents the cost associated with allowing these discontinuities in the warping path.

Note that the use of the null phonemes at the beginning and end of each string does not change the unconstrained endpoint nature of the optimization. The optimal path, while always passing through the final null node, can "jump" from any permissible previous node to the final null node. After the last nonnull phoneme has been processed in each string, the optimal path has been determined. The final null phonemes are introduced for convenience in that the optimal path is easily located using these artificial nodes.

A key component in assuring acceptable performance of the matching algorithm is the construction of an accurate phoneme-to-phoneme distance matrix. The distance matrix, which is symmetric, is shown in Table II. The basic characteristic of the distance matrix is that, with a few exceptions, vowels are not permitted to match with consonants. The various vowel-to-vowel distances and consonant-to-consonant distances are somewhat arbitrary, and attempt to approximate data in [1], [5], and [6].

An important consideration in this algorithm is the cost associated with allowing an insertion or rejection error (a match of either a stimulus or response phoneme to null) as opposed to the cost in allowing a substitution error. The interactions between these two weights, to some degree, control the number of insertion and rejection errors in the final alignment. For example, the consonant "K" is prevented from matching to the diphthong "OY" as a substitution error by defining the cost associated with that match to be higher than the cost associated with matching "K" to the null phoneme "**."

In this system, alignments which violate word boundaries are generally not penalized, except for one special case. Consider the initial "T" phoneme in the stimulus string of Fig. 1. While it can arguably be matched to either "T" in the response string, the human expert will invariably construct the alignment such that the word "test" in the stimulus string matches identically with the word "test" in the response string, as shown in Fig. 2. In this special case, where a phoneme can occur in several different places with equal cost, the alignment which violates the least word boundaries is chosen. Although this has negligible impact at the phoneme level, this will assure an alignment at the word level which is closer to that produced by the human expert.

The result of this stage of processing is a phoneme-to-phoneme matching from which substitutions, insertions, and rejection rates at the phoneme level can be computed. The algorithm output for the warping path of Fig. 1 is shown in Fig. 2. By tracking the location of word boundaries, it is possible to transform the phonemically aligned transcriptions into word-aligned strings, as shown in Fig. 2. This word alignment is merely a reflection of the underlying phonemic matching. There is no attempt made to reconcile word boundaries. From this word-level alignment, word error rates (as referenced to the stimulus string) can also be computed in terms of substitution, insertion, and rejection errors.

## III. Evaluation

The most meaningful way to benchmark the accuracy of this system is to compare the algorithms' output to that of the human expert. One list of 50 phrase pairs was prepared and analyzed by three "expert speech scientists." These data represent a subset of some data collected from an in-house speech intelligibility test measuring the performance of an LPC vocoder and from an in-house text-to-speech synthesis system evaluation. (In each of these tests, listeners were presented with some audio stimulus, and asked to transcribe its contents.) For the benchmark evaluation, a set of 50 phrase pairs was presented to each of three experts in the format shown in Fig. 3. Each expert was asked to align the text at both the word level and the phoneme level, and then tabulate the results in terms of substitution, insertion, and rejection errors.

The results of this benchmark are shown in Table III, for both the word-level alignments and the phoneme-level alignments. The data for each expert are listed separately, followed by the cumulative statistics in terms of the averages and standard deviations for each error category. Finally, the results for the automatic algorithm are listed. Note that the total number of errors per category need not be the same for all experts. The major source of variation is due to the individuals' criterion as to what constitutes a substitution error, as opposed to what constitutes a rejection or insertion error, that is, which phoneme confusions (or word confusions) are permitted.

A more revealing statistic involves the subset of the data which were scored identically by all three experts. Of these 50 phrase pairs presented to the experts, the experts' responses were identical for 35 phrase pairs. The algorithm produced alignments which were identical to the experts' alignments on all 35 of these phrase pairs. Both these benchmarks indicate that the algorithms' performance is quite close to that of the human expert.

TABLE II
(a) VOWEL–VOWEL DISTANCE MATRIX. (b) CONSONANT–CONSONANT DISTANCE MATRIX. (c) VOWEL–CONSONANT DISTANCE MATRIX

| | IY | UW | EY | OW | AE | AA | AY | AW | OY | AO | IH | UH | EH | AX | ER | Y | W | ** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IY | 0 | 95 | 80 | 95 | 90 | 95 | 80 | 75 | 95 | 90 | 60 | 90 | 80 | 95 | 0 | 95 | 95 | 100 |
| UW | 95 | 0 | 95 | 80 | 95 | 95 | 90 | 80 | 95 | 95 | 90 | 60 | 95 | 90 | 95 | 95 | 95 | 100 |
| EY | 80 | 95 | 0 | 90 | 80 | 95 | 95 | 95 | 90 | 90 | 80 | 95 | 60 | 60 | 80 | 95 | 95 | 100 |
| OW | 95 | 80 | 90 | 0 | 95 | 90 | 95 | 95 | 60 | 80 | 75 | 80 | 90 | 70 | 80 | 95 | 95 | 100 |
| AE | 90 | 95 | 80 | 95 | 0 | 79 | 60 | 95 | 70 | 90 | 60 | 95 | 80 | 60 | 90 | 95 | 95 | 100 |
| AA | 95 | 95 | 95 | 90 | 79 | 0 | 80 | 90 | 90 | 70 | 95 | 95 | 95 | 80 | 95 | 95 | 95 | 100 |
| AY | 80 | 90 | 95 | 95 | 60 | 80 | 0 | 70 | 95 | 70 | 95 | 90 | 95 | 80 | 95 | 95 | 95 | 100 |
| AW | 75 | 80 | 95 | 95 | 95 | 90 | 70 | 0 | 95 | 80 | 95 | 80 | 95 | 90 | 95 | 95 | 95 | 100 |
| OY | 95 | 95 | 90 | 60 | 70 | 90 | 95 | 95 | 0 | 80 | 95 | 95 | 90 | 90 | 95 | 95 | 95 | 100 |
| AO | 90 | 95 | 90 | 80 | 90 | 70 | 70 | 80 | 80 | 0 | 95 | 95 | 95 | 90 | 90 | 95 | 95 | 100 |
| IH | 60 | 90 | 80 | 75 | 60 | 95 | 95 | 95 | 95 | 95 | 0 | 90 | 80 | 90 | 60 | 95 | 95 | 100 |
| UH | 90 | 60 | 95 | 80 | 95 | 95 | 90 | 80 | 95 | 95 | 90 | 0 | 95 | 90 | 64 | 95 | 95 | 100 |
| EH | 80 | 95 | 60 | 90 | 80 | 95 | 95 | 95 | 90 | 95 | 80 | 95 | 0 | 80 | 80 | 95 | 95 | 100 |
| AX | 95 | 90 | 60 | 70 | 60 | 80 | 80 | 90 | 90 | 90 | 90 | 90 | 80 | 0 | 70 | 95 | 95 | 100 |
| ER | 0 | 95 | 80 | 80 | 90 | 95 | 95 | 95 | 95 | 90 | 60 | 64 | 80 | 70 | 0 | 95 | 95 | 100 |
| Y | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 0 | 60 | 100 | |
| W | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 60 | 0 | 100 | |

| | R | L | P | B | F | V | M | T | D | DX | TH | DH | N | S | Z | CH | JH | SH | ZH | K | G | NX | HH | WH | ** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | 0 | 80 | 90 | 90 | 90 | 80 | 70 | 95 | 95 | 95 | 95 | 90 | 90 | 90 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 95 | 100 |
| L | 80 | 0 | 90 | 90 | 90 | 80 | 60 | 61 | 95 | 95 | 95 | 90 | 90 | 95 | 95 | 95 | 95 | 95 | 95 | 85 | 95 | 95 | 95 | 95 | 100 |
| P | 90 | 90 | 0 | 60 | 60 | 60 | 70 | 60 | 60 | 60 | 70 | 70 | 70 | 90 | 90 | 80 | 80 | 90 | 85 | 70 | 70 | 70 | 70 | 70 | 100 |
| B | 90 | 90 | 60 | 0 | 70 | 60 | 60 | 60 | 60 | 60 | 70 | 70 | 80 | 90 | 90 | 80 | 90 | 90 | 90 | 70 | 60 | 80 | 60 | 60 | 100 |
| F | 90 | 90 | 60 | 70 | 0 | 60 | 90 | 70 | 80 | 80 | 60 | 60 | 80 | 60 | 70 | 90 | 60 | 60 | 60 | 60 | 80 | 90 | 60 | 60 | 100 |
| V | 80 | 80 | 60 | 60 | 60 | 0 | 60 | 70 | 80 | 90 | 60 | 70 | 65 | 80 | 70 | 95 | 80 | 95 | 85 | 95 | 80 | 95 | 75 | 70 | 100 |
| M | 70 | 60 | 70 | 60 | 90 | 60 | 0 | 75 | 90 | 90 | 90 | 70 | 60 | 90 | 80 | 90 | 70 | 90 | 85 | 90 | 80 | 90 | 90 | 90 | 100 |
| T | 95 | 61 | 60 | 60 | 70 | 70 | 75 | 0 | 60 | 60 | 60 | 60 | 90 | 61 | 90 | 60 | 70 | 60 | 90 | 60 | 70 | 70 | 95 | 95 | 100 |
| D | 95 | 95 | 60 | 60 | 80 | 80 | 90 | 60 | 0 | 60 | 60 | 60 | 80 | 90 | 95 | 70 | 85 | 95 | 95 | 60 | 60 | 60 | 95 | 95 | 100 |
| DX | 95 | 95 | 60 | 60 | 80 | 90 | 90 | 60 | 60 | 0 | 60 | 60 | 90 | 90 | 95 | 70 | 85 | 95 | 95 | 70 | 70 | 70 | 90 | 95 | 100 |
| TH | 95 | 95 | 70 | 70 | 60 | 60 | 90 | 60 | 60 | 60 | 60 | 60 | 90 | 60 | 70 | 85 | 85 | 60 | 75 | 80 | 70 | 80 | 60 | 80 | 100 |
| DH | 90 | 90 | 70 | 70 | 60 | 70 | 70 | 60 | 60 | 60 | 60 | 0 | 90 | 90 | 90 | 75 | 75 | 75 | 75 | 60 | 65 | 65 | 80 | 80 | 100 |
| N | 90 | 90 | 70 | 80 | 80 | 65 | 60 | 90 | 80 | 90 | 90 | 90 | 0 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 60 | 90 | 90 | 100 |
| S | 90 | 95 | 90 | 90 | 60 | 80 | 90 | 61 | 90 | 90 | 60 | 90 | 90 | 0 | 65 | 90 | 80 | 60 | 70 | 60 | 95 | 95 | 95 | 95 | 100 |
| Z | 95 | 95 | 90 | 90 | 70 | 70 | 80 | 90 | 95 | 95 | 70 | 90 | 90 | 65 | 0 | 80 | 60 | 60 | 60 | 95 | 95 | 70 | 70 | 90 | 100 |
| CH | 95 | 95 | 80 | 80 | 90 | 95 | 90 | 60 | 70 | 70 | 85 | 75 | 90 | 90 | 80 | 0 | 80 | 80 | 80 | 60 | 85 | 90 | 90 | 95 | 100 |
| JH | 95 | 95 | 80 | 90 | 60 | 80 | 70 | 70 | 85 | 85 | 85 | 75 | 90 | 80 | 60 | 80 | 0 | 70 | 60 | 80 | 80 | 90 | 95 | 90 | 100 |
| SH | 95 | 95 | 90 | 90 | 60 | 95 | 90 | 60 | 95 | 95 | 60 | 75 | 90 | 60 | 60 | 80 | 70 | 0 | 60 | 95 | 95 | 95 | 85 | 85 | 100 |
| ZH | 95 | 95 | 85 | 90 | 60 | 85 | 85 | 90 | 95 | 95 | 75 | 75 | 90 | 70 | 60 | 80 | 80 | 60 | 0 | 95 | 95 | 95 | 95 | 95 | 100 |
| K | 95 | 65 | 70 | 70 | 80 | 95 | 90 | 60 | 60 | 70 | 80 | 60 | 90 | 60 | 95 | 60 | 80 | 95 | 95 | 0 | 80 | 90 | 95 | 95 | 100 |
| G | 95 | 95 | 70 | 60 | 90 | 80 | 80 | 70 | 60 | 70 | 70 | 65 | 90 | 95 | 95 | 65 | 80 | 95 | 95 | 60 | 0 | 60 | 95 | 95 | 100 |
| NX | 95 | 95 | 70 | 80 | 90 | 95 | 90 | 70 | 60 | 70 | 80 | 65 | 60 | 95 | 70 | 90 | 90 | 95 | 95 | 90 | 60 | 0 | 95 | 95 | 100 |
| HH | 95 | 95 | 70 | 60 | 60 | 75 | 90 | 95 | 95 | 90 | 60 | 80 | 90 | 95 | 70 | 90 | 95 | 85 | 95 | 95 | 95 | 95 | 0 | 80 | 100 |
| WH | 95 | 95 | 70 | 60 | 60 | 70 | 90 | 95 | 95 | 95 | 80 | 80 | 90 | 95 | 90 | 95 | 90 | 85 | 95 | 95 | 95 | 95 | 80 | 0 | 100 |

| | R | L | P | B | F | V | M | T | D | DX | TH | DH | N | S | Z | CH | JH | SH | ZH | K | G | NX | HH | WH | ** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IY | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| UW | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| EY | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| OW | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| AE | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| AA | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| AY | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| AW | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| OY | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| AO | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| IH | 95 | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| UH | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| EH | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| AX | 95 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| ER | 90 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 100 |
| Y | 65 | 70 | 90 | 70 | 60 | 60 | 90 | 95 | 95 | 90 | 90 | 90 | 90 | 90 | 90 | 85 | 70 | 96 | 85 | 85 | 85 | 85 | 85 | 70 | 100 |
| W | 70 | 70 | 80 | 70 | 60 | 55 | 95 | 95 | 90 | 95 | 90 | 90 | 90 | 95 | 85 | 95 | 80 | 90 | 90 | 95 | 95 | 95 | 90 | 60 | 100 |

```
Input Response Word(s):  the best test
Input Stimulus Word(s):  a test

Output Response Word(s): TH-e BEST test
Output Stimulus Word(s): ** a **** test

   ( 1) Sub. ( 1) Ins. ( 0) Rej.  ( 2) Total Errors

Output Response Phonemes: DH AX B  EH S  T  T EH S T
Output Stimulus Phonemes: ** AX ** ** ** ** T EH S T

   ( 0) Sub. ( 5) Ins. ( 0) Rej.  ( 5) Total Errors
```

Fig. 2. Typical output displaying alignments and error tabulation.

## IV. CONCLUSIONS

This two-stage process of aligning strings of words has been found to perform quite acceptably for applications ranging from listening tests to speech recognition system evaluation. We find the ability to study phoneme confusion matrices generated from the experimental data an invaluable tool in speech algorithm development. The weakest link in the present algorithm is the construction of the static phoneme distance matrix. The performance of the system is heavily dependent upon the accuracy of this distance matrix. An unresolved issue, at this time, is the importance of context dependent distances in arriving at the "optimal" match (as defined by the human expert). As we acquire more data about phoneme-to-phoneme confusions, as judged by the expert phonetician, the phoneme confusion matrix can be refined, and the cost function modified, such that algorithm performance is optimized. The present performance of the system, however,

```
Data Presented To Expert:

Input Response:  TO WRECK A NICE BEACH
Input Stimulus:  TO RECOGNIZE SPEECH

words    : (   ) sub. (   ) ins. (   ) rej.

Input Response: T UW R EH K AX N AY S B IY CH
Input Stimulus: T UW R EH K AX G N AY Z S P IY CH

phonemes: (   ) sub. (   ) ins. (   ) rej.

Expert's Response:

Input Response:  TO WRECK A * NI---CE BEACH
Input Stimulus:  TO REC---O-G-NIZE S--PEECH

words    : ( 2) sub. ( 0) ins. ( 0) rej.

Input Response: T UW R EH K AX ** N AY ** S B IY CH
Input Stimulus: T UW R EH K AX G  N AY Z  S P IY CH

phonemes: ( 1) sub. ( 0) ins. ( 2) rej.
```

Fig. 3. An example of the evaluation data.

TABLE III
(a) BENCHMARK EVALUATION OF WORD-LEVEL ACCURACY. (b) BENCHMARK EVALUATION OF PHONEME-LEVEL ACCURACY

Cumulative Word-Level Results
(50 Phrase Pairs, 210 Words Total)

| Expert No. | Subst. | Ins. | Rej. | Total Errors |
|---|---|---|---|---|
| 1 | 114 | 14 | 14 | 142 |
| 2 | 115 | 14 | 12 | 141 |
| 3 | 118 | 10 | 10 | 138 |
| Avg.: | 115.7 | 12.7 | 12.0 | 140.3 |
| St.Dev.: | 1.7 | 1.9 | 1.6 | 1.7 |
| Algorithm: | 116 | 13 | 12 | 141 |

Cumulative Phoneme-Level Results
(50 Phrase Pairs, 820 Phonemes Total)

| Expert No. | Subst. | Ins. | Rej. | Total Errors |
|---|---|---|---|---|
| 1 | 140 | 87 | 109 | 336 |
| 2 | 158 | 90 | 84 | 332 |
| 3 | 146 | 91 | 98 | 335 |
| Avg.: | 148.0 | 89.3 | 97.0 | 334.3 |
| St.Dev.: | 7.5 | 1.7 | 10.2 | 1.7 |
| Algorithm: | 145 | 89 | 96 | 330 |

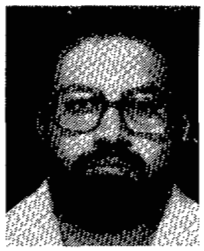
has proven to be quite acceptable for a variety of applications.

## ACKNOWLEDGMENT

The authors would like to thank R. Schwartz of Bolt, Beranek, and Newman, Inc., for his helpful suggestions on techniques to reduce the computation required to execute the dynamic programming optimization.

## REFERENCES

[1] A. S. House, C. E. Hecker, and K. D. Kryter, "Articulation-testing methods: Consonantal differentiation with a closed-response set," *J. Acoust. Soc. Amer.* vol. 37, pp. 158–166, Jan. 1965.
[2] "IEEE recommended practice for speech quality measurements," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 225–246, Sept. 1969.
[3] K. M. Goudie, "Defining and developing intelligibility in a text to speech system," *J. Acoust. Soc. Amer.*, suppl. 1, vol. 76, no. S1, p. S2, Fall 1984.
[4] W. D. Voiers, "Diagnostic evaluation of speech intelligibility," in *Speech Intelligibility and Speaker Recognition*, M. E. Hawley, Ed. Stroudsburg, PA: Dowden, Hutchinson, and Ross, 1977.
[5] G. E. Peterson and H. L. Barney, "Control methods used in a study of the vowels," *J. Acoust. Soc. Amer.*, vol. 24, pp. 175–184, Mar. 1952.
[6] G. A. Miller and P. A. Nicely, "An analysis of perceptual confusions among some English consonants," in *Readings in Acoustic Phonetics*, I. Lehiste, Ed. Cambridge, MA: M.I.T. Press, 1967, pp. 301–315.
[7] W. M. Fisher, "A text-to-speech development system, in *Proc. 1983 Int. Conf. Acoust., Speech, Signal Processing.*, Apr. 1983, pp. 1344–1347.



**Joseph Picone** (S'79–M'83) received the B.S.E.E. degree in 1979, the M.S.E.E. degree in 1980, and the Ph.D. degree in 1983, all from the Illinois Institute of Technology, Chicago.

During 1981–1983 he was involved in research into medium data rate speech coding at Bell Laboratories in Naperville, IL. From 1983 to 1985 he was with the Speech Systems Research Group at Texas Instruments, Inc., where he was engaged in research into low data rate speech coding. He is currently with the Exploratory Voice Capabilities Department, AT&T Bell Laboratories, Naperville, where he is involved in research into speech coding.

**Kathleen M. Goudie-Marshall** was born in Detroit, MI, on November 2, 1950. She received the B.A. degree in psychology and linguistics from Oakland University in Rochester, MI, in 1971, and the M.A. and the Ph.D. degrees in linguistics from the University of Michigan, Ann Arbor, in 1976 and 1979, respectively.

From 1977 to 1978 she assisted in the teaching of phonetics, after which she completed her doctoral research in intonation and discourse. From October 1979 to December 1983 she worked in the Consumer Products Group of Texas Instruments, Inc., doing synthetic speech research, and transferred to the Corporate Research Design and Engineering Division into the speech research group under G. Doddington, where she has been working on text-to-speech synthesis by rule and speech recognition research.

**George R. Doddington** (M'79), for a photograph and biography, see this issue, p. 764.

**William Fisher** (M'78) received the B.S. degree in mechanical engineering and the M.S. degree in engineering from Purdue University, West Lafayette, IN, and the M.A. and Ph.D. degrees in linguistics from the University of Chicago, Chicago, IL, where his thesis work involved him in a computerized field study of Mayan dialects.

He has worked for R. R. Donnelley and Sons, Inc., as a Digital Systems Analyst, Engineer, and Project Leader, taught phonology and phonetics at Washington University, and was previously employed in speech research at the Central Institute for the Deaf in St. Louis, MO. He is currently a member of the Technical Staff of Texas Instruments, working in the Central Research Laboratories, where he has done research in medium-cost text-to-speech systems, grammar-controlled speech recognition, and the design of speech databases. His interest in language and applications of linguistics lured him away from a career as an Aerospace Engineer.

Dr. Fisher is a member of Sigma Xi, the Association for Computing Machinery, and the Association for Computational Linguistics.