(4)

BBN Laboratories Incorporated

A Subsidiary of Bolt Beranek and Newman Inc.

BBN Report No. 6725

STATISTICAL MODELING FOR CONTINUOUS SPEECH RECOGNITION

Final Report

R. Schwartz, Y-L. Chow, A. Derr, M-W. Feng, O. Kimball,
F. Kubala, J. Makhoul, M. Ostendorf, P. Price, S. Roucos

February 1988

BBN Report No. 6725

ARPA Order Number 4707
Contract Number N00014-85-C-0279
Contract Duration: 17 Jan 1985 - 16 Jan 1988
Principal Investigator: J. Makhoul, (617)873-3332

Final Report

STATISTICAL MODELING FOR CONTINUOUS SPEECH RECOGNITION

R. Schwartz, Y-L. Chow, A. Derr, M-W. Feng, O.Kimball,
F. Kubala, J. Makhoul, M. Ostendorf, P. Price, S. Roucos

February 1988

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-A192 054

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER BBN Report No. 6725	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Statistical Modeling for Continuous Speech Recognition		5. TYPE OF REPORT & PERIOD COVERED Final Report Jan. 1985 - Jan. 1988
		6. PERFORMING ORG. REPORT NUMBER BBN Report No. 6725
7. AUTHOR(s) R. Schwartz, Y-L. Chow, A. Derr, M-W. Feng, O. Kimball, F. Kubala, J. Makhoul, M. Ostendorf, P. Price, S. Roucos		8. CONTRACT OR GRANT NUMBER(s) N00014-85-C-0279
9. PERFORMING ORGANIZATION NAME AND ADDRESS BBN Laboratories 10 Moulton Street Cambridge, MA 02238		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, Virginia 22217-5000		12. REPORT DATE February 1988
		13. NUMBER OF PAGES 46
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)

Distribution of the document is unlimited. It may be released to the Clearinghouse, Dept. of Commerce, for sale to the general public.

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

18. SUPPLEMENTARY NOTES

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

speech recognition, phonetic recognition, continuous speech, hidden Markov model, coarticulation, statistical segment modeling, speaker adaptation

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

~~Our~~ The authors' research into developing robust, high-performance continuous speech recognition systems for large-vocabulary tasks, such as battle management, has focused on the development of accurate mathematical models for the different phonemes that occur in English. The research performed in this project has been in three general areas: Hidden Markov Models, Stochastic Segment Models, and Rapid Speaker Adaptation.

over

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Hidden Markov models and stochastic segment models are two distinct methods of modeling phonetic coarticulation, i.e., the variation of phonemes in the context of other phonemes. The authors We have tested the use of context-dependent hidden Markov models in BYBLOS, the BBN continuous speech recognition system, and we report on word recognition accuracy in a 1000-word task domain. In contrast to hidden Markov modeling which models each part of a phoneme independently, stochastic segment modeling models each phoneme as a whole unit, and therefore has the promise of improved performance, as our preliminary experiments indicate.

Most of the work reported has been performed in speaker-dependent mode, which utilizes 300 to 600 sentences for training the system on a speaker's voice. In an effort to minimize the amount of training for a new speaker, we have initiated an effort to develop speaker adaptation methods that require only 10 to 40 sentences from the new speaker. Initial results in speaker adaptation show that the word recognition error rate is only about twice the error rate in speaker-dependent mode.

Table of Contents

1. Executive Summary 1

2. Robust Model of Phonetic Coarticulation 3

2.1 Model of Phonetic Coarticulation 3
2.1.1 Training Problem 7
2.1.2 Combined Model 7
2.2 Experiments with Combined Context Model 8
2.2.1 E-set Problem 9
2.2.2 Continuous Phonetic Recognition 9
2.2.3 Coarticulatory Effects in a Word Recognition System 10
2.2.4 Recognition With a Grammar 15
2.2.5 Conclusion 15
2.3 BYBLOS System Recognition Results 16
2.4 Summary 18

3. Stochastic Segment Modeling 19

3.1 Stochastic Segment Model 19
3.1.1 Resampling Transformations 20
3.1.2 Probabilistic Model 21
3.1.3 Properties of the Segment Model 21
3.2 Recognition Algorithm 22
3.3 Training Algorithm 23
3.4 Experimental Results 24
3.4.1 Phoneme Recognition 24
3.4.2 Word Recognition 25
3.5 Conclusion 27

4. Rapid Speaker Adaptation 28

4.1 Probabilistic Mapping 28
4.2 Estimation of Transformation Matrix 31
4.2.1 Method 31
4.2.2 Experiments 32
4.2.3 Results 33
4.3 Estimation of Matrix using Text-Dependent Alignment 36
4.3.1 Method 36

4.3.2 Experiments 37
4.3.3 Results 38
4.3.4 Conclusions 41

List of Figures

Figure 1: Several Different Levels of Representation for Speech. The phrase is "grey whales". 4

Figure 2: Hid^den Markov Model of a Phoneme. States 1, 2, and 3 are assumed to correspond approximately to the left, middle, and right portion of a phoneme. 6

Figure 3: Word Recognition Accuracy using different coarticulation models. 14

Figure 4: Observed input segment (o) and corresponding resampled segment (x). The two axes correspond to two cepstral coefficients. 20

Figure 5: Expanded HMM. a) single state of the HMM; b) expanded model separating prototype pdf and transformation matrix. 30

Figure 6: Speaker-Dependent Training vs adaptation. Speaker-Dependent Training (○); Speaker adaptation (□); Cross-Speaker Results (△). The solid line indicates accuracy with a grammar; the dashed line indicates no grammar. 34

Figure 7: Comparison of Different Methods for Estimating the Transformation Matrix 39

List of Tables

Table 1: Different System Configurations for Word Recognition. 13

Table 2: Recognition results with a grammar. 15

Table 3: Continuous speech recognition results with grammars. 17

Table 4: Recognition results using manually segmented speech and automatically segmented speech. 25

Table 5: Word recognition/insertion rates for three speakers for the segment phoneme system and for two HMM systems: phoneme models and phoneme models conditioned on the left and right context. 26

Table 6: Source of Recognition Errors. Each line changes one experimental condition. 35

Table 7: Comparison of Recognition Accuracy of Speaker Adaptation using 2 minutes vs Speaker-Dependent Training using 30 minutes 41

1. Executive Summary

This report describes the research in continuous speech recognition performed under Contract No. N00014-85-C-0279. The contract period ran from January 17, 1985 through January 16, 1988.

The general goal of this work has been to develop methods for improved performance in large vocabulary continuous speech recognition. Our research into developing robust, high-performance continuous speech recognition systems for large-vocabulary complex tasks, such as battle management, has focused on the development of accurate mathematical models for the different phonemes that occur in English. The research performed in this project has been in three general areas: Hidden Markov Models, Stochastic Segment Modeling, and Rapid Speaker Adaptation algorithms.

It is well-known that the acoustic realization of phonemes are affected significantly by the surrounding phonetic context. This effect is known as *coarticulation*. Attempts to model these effects in the past have mainly centered on using speech units larger than the phoneme, such as the syllable or word. In this work, we developed a new technique for modeling context-dependent phonetic units of speech, which allows more accurate modeling of the coarticulation effects in speech. The work is based on robust Hidden Markov Models (HMM) of phonemes in context. That is, the method allows the statistical models of the phonemes to be conditioned on any degree of context that is most useful. In particular, we have used models conditioned on the immediate phonetic context as well as the word in which the phoneme appears. The results of this work in phonetic modeling has been incorporated in BYBLOS, the BBN continuous speech recognition system, which has demonstrated word recognition accuracy of 98.5% for a 1000-word task and a perplexity (branching factor) of 10, a word accuracy of 93% for a perplexity of 50, and 75% for a perplexity of 1000.

In addition to our work on HMM models of speech, we have developed a new model called the Stochastic Segment Model, which models each phoneme as a whole unit (in contrast to the HMM model, which models each part of the phoneme independently). We expect the segment model to result in a more accurate representation of the time variations in speech. Our preliminary work on the segment model shows that in some cases it achieves significantly better recognition accuracy than the HMM model.

Most of our work had been in speaker-dependent mode, which utilizes 300 to 600 sentences for training the system on a speaker's voice (equivalent to 15 to 30 minutes of continuous speech). In an effort to minimize the amount of training for a new speaker, we have started work on developing speaker adaptation methods that require only 10 to 40 sentences from the new speaker. The method that we have developed uses a few sentences to transform a well-trained HMM model from a single prototype speaker so that it can model the speech of the new speaker. We have shown that, using this new method, the word recognition error rate is about twice the error rate in speaker-dependent mode.

In Section 2 of this report we describe our work on a Robust Model of Phonetic Coarticulation, and provide results for this model within the BYBLOS system. We present our work on the Stochastic Segment model in Section 3, and in Section 4, we describe our work in Rapid Speaker Adaptation.

2. Robust Model of Phonetic Coarticulation

The largest area of our work to date is our work on Hidden Markov models. We have developed a comprehensive mathematical model of phonetic coarticulation in continuous speech. In this section we provide the motivation for a coarticulation model, define a particular model that seems to fit what we know of coarticulation, and present the results of several speech recognition experiments that show the model to be quite useful.

2.1 Model of Phonetic Coarticulation

Hidden Markov Models (HMMs) have been shown to provide an effective statistical formalism for speech recognition. They have been used to model whole words in both isolated [1] and continuous [2] speech recognition. They have also been used to model phonemes for continuous speech recognition [2, 3, 4]. The Hidden Markov Model has two important advantages over many other models for speech. First, it provides a well-defined structural model for variability in both time and in frequency (spectral variation), both of which occur in speech. Second, once the structure of the models are specified, the parameters of the models can be estimated automatically with a large amount of speech data using the forward-backward or Baum-Welch algorithm [5].

It is generally assumed that large-vocabulary continuous speech recognition systems should be phonetically based. That is, each word in the lexicon is decomposed into phoneme subunits, each of which is modeled separately. The use of a phonetic model makes it easy to model phonological variation both within and across words. It also makes it possible for a new speaker to use the system without first saying all the words in the lexicon.

It is well known that phonemes are affected significantly by adjacent phonemes in a process known as coarticulation. To obtain high recognition accuracy it is important to model phonetic coarticulation as well as possible. To explain our model for coarticulation, we must first define some terms.

Figure 1 illustrates several different levels of representation for speech. The figure illustrates the levels of words, phonemes, allophones, allophone models, and analyzed speech parameters. The phrase shown is "grey whales". The purpose of speech recognition is to determine the sequence of words corresponding to an observed utterance. We often decompose words into sequences of basic speech sounds or phonemes, to try to reduce the problem of modelling many words to the problem of modelling a smaller number of units. We observe that

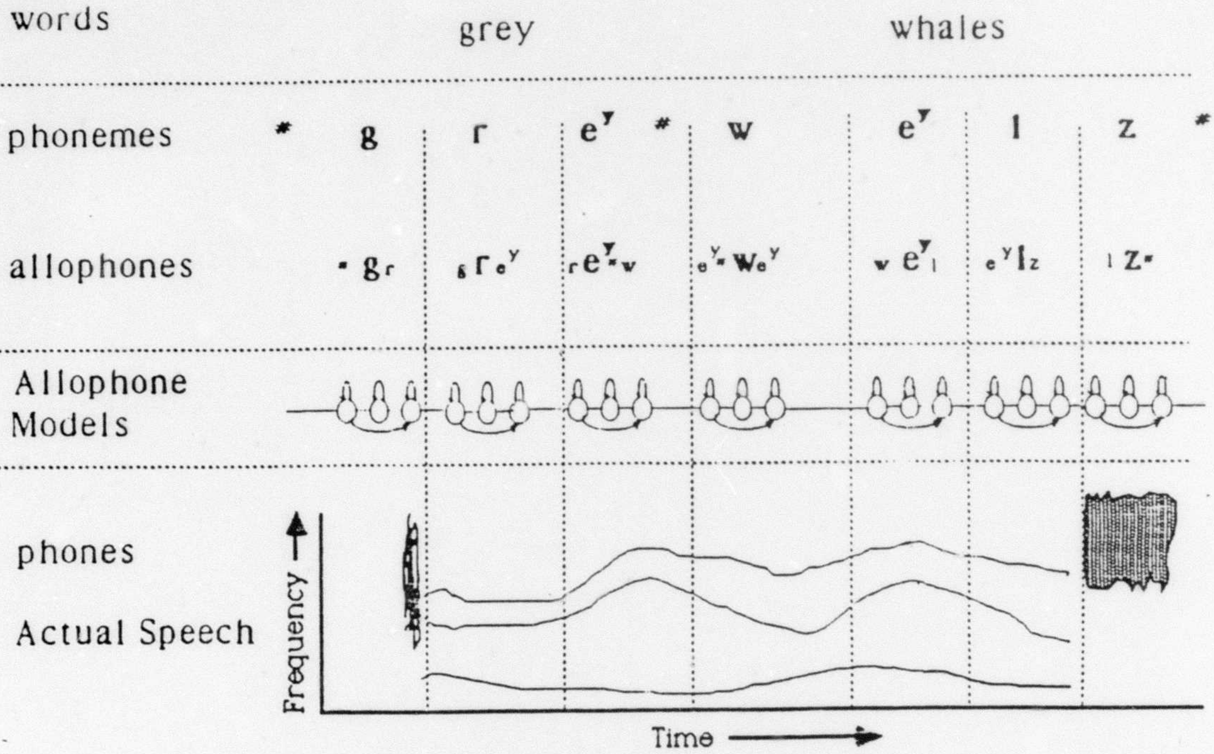

Figure 1: Several Different Levels of Representation for Speech. The phrase is "grey whales".

these basic units exhibit systematic acoustic variation as a function of their phonetic environment. To capture this systematic variation we must first define context-dependent allophones or variants of each phoneme. An allophone is defined as any variant of a phoneme, which may be statistically different from other allophones of that phoneme. We have shown allophones defined by the preceding and following contexts. We will often use the terms "left" and "right" instead of "preceding" and "following". At the bottom of the figure is shown a schematic of the formant tracks corresponding to a single utterance of the phrase. This (or any other) parametric representation of the spoken speech will be different for each utterance of the phrase. Therefore, we need statistical models to represent the likely acoustic realizations (phones) for each allophone. While many different statistical models are available, we have chosen to use hidden Markov models as our basic allophone model for the reasons given previously.

As illustrated in the figure, the coarticulation effects bridge all the phoneme boundaries. If we allow the coarticulatory dependency of each speech unit to extend beyond its duration, then we can model any amount of dependency that we wish. For example, in the illustration shown, we have modeled the effect of each phoneme on its immediate neighbors. This idea of *context-dependent* units is key to the modeling of coarticulation.

Figure 2 illustrates the HMM that we use to model a phoneme. The circles represent states of the model. We define s_t to be the state of the Markov process at time t. At each time, t, we also have an observed spectral envelope model, expressed as a vector, $\underline{x}_t$. With each state, i, is associated a probability distribution function (pdf)

$$b_i(\underline{x}) = p(\underline{x}_t | s_t = i); \qquad i=1,2,3 \qquad (1)$$

for the observed spectral vector, $\underline{x}_t$, given that the process is in state i at time t. Since the process is Markov, the pdfs do not depend on t. In our implementation, we use discrete pdfs for the vector $\underline{x}$. First, a portion of the training speech for a speaker is analyzed and then used to determine a codebook of spectral templates using a clustering algorithm [6]. Then, for any spectral envelope model vector, $\underline{x}$, using vector quantization (VQ) we search the codebook for the template vector that is closest. The index of the closest vector, v_t, then defines a bin of a discrete probability distribution. The pdfs in our HMMs, then, have a probability for each of the possible bins:

$$b_i(v) = p(v_t = k | s_t = i); \qquad i=1,2,3; \qquad for \;\; all \;\; k \qquad (2)$$

For each allowed transition (indicated by the arrows in Figure 2) we have a transition probability

$$a_{ij} = P\,(s_t = j | s_{t-1} = i) \qquad (3)$$

the probability of state i being followed by state j. While the relation is not direct, we find it useful to think of the states as corresponding to the beginning, middle, and end of a phoneme.

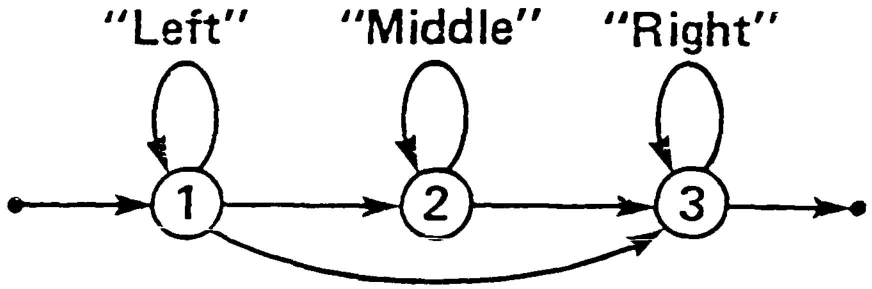

Figure 2: Hidden Markov Model of a Phoneme. States 1, 2, and 3 are assumed to correspond approximately to the left, middle, and right portion of a phoneme.

Next, we discuss the issues of training set size and robustness that arise with the use of large numbers of models.

2.1.1 Training Problem

For any units with context-dependency larger than the phoneme, we will have a training problem. While some of the contexts may occur frequently, many will not occur with sufficient frequency to estimate a robust acoustic model. In fact, large numbers of the possible contexts will not occur at all in any particular set of training speech.

A simple solution would be to use the most detailed context-dependent model with a sufficient number of training samples. For example, let us say we want a model for the /e^y/ in "whales". If the word whales has appeared a few times, we would use the model of /e^y/ that depended on the word whales. If not, we might use a model of all /e^y/ that are preceded by /w/ and followed by /l/ (as in "away late"). If this context did not occur, then we could fall back to a model dependent on the left or right context alone (as in "wait" or "tail"). Or if nothing else, we could resort to the context-independent model derived from all /e^y/ phoneme tokens. This algorithm for choosing the model, however, does not make optimal use of the training data, and does not properly account for coarticulatory phenomena. To solve this problem, we must examine more closely how coarticulation interacts with our model for a phoneme.

2.1.2 Combined Model

Both experience and reason tell us that the coarticulatory effect of an adjacent phoneme is greatest in the part of the phoneme closest to that adjacent phoneme. For example, a phoneme to the left will have the most effect on the left part (state 1) of a phoneme, and the least effect on the right part (state 3). To account for both the nature of coarticulation and the requirements for robust statistical models, we use a combined context model as shown in the following example:

Example:

Model for /e^y/ in "whales"

$$
\begin{aligned}
\hat{p}(\underline{y} | e^y \text{ in whales}) = {} & \lambda_1 \; p(\underline{y} | w \; e^y \; l) \\
& + \lambda_2 \; p(\underline{y} | w \; e^y \;) \\
& + \lambda_3 \; p(\underline{y} | \; e^y \; l) \\
& + \lambda_4 \; p(\underline{y} | \; e^y \;) \\
& + \lambda_5 \; p(\underline{y} | \; e^y \text{ in whales})
\end{aligned}
$$

$$\underline{\lambda} = f(\#\ \text{Occurrences, State})$$

$$\sum_{k=1}^{5} \lambda_k = 1$$

That is, the combined model, $\hat{p}$, is a linear combination of the various context-dependent models. The weight vector, $\underline{\lambda}$, depends on the state of the phoneme model (left, middle, right), and the amount of training for each model. The sum of the weights, λ, from any node is 1. During forward-backward training the models are kept separate. Prior to recognition, the models for a state can be combined into a single pdf to save computation. Thus, during recognition, the HMM is of the same complexity as a single unconditioned model.

To summarize, we have argued that we can model coarticulation effects by the use of context-dependent models of phonemes. Furthermore, to avoid the lack of robustness due to insufficient amounts of training, we can smooth these detailed context-dependent models with well trained context-independent models. The amount of the smoothing depends on both the location of the HMM state in the phoneme, and the amount of training available for that particular context.

2.2 Experiments with Combined Context Model

Below we describe a succession of experiments designed to demonstrate the effectiveness of the coarticulation model proposed above.

2.2.1 E-set Problem

The "E-set" is the set of nine letters of the English alphabet that rhyme with E. They are B, C, D, E, G, P, T, V, Z. They provide a few interesting problems for speech recognition. First, since they differ phonetically in only one phoneme, they require minimal pair distinctions. Second, since most of the duration of each utterance is the /i/ phoneme, one has to be careful that random statistical variation in this region does not dominate in the total discrimination score. Third, the models for the consonants do not depend on phonetic context, since they always appear preceded by silence (in isolated speech), and followed by /i/. The /i/ phoneme, however, appears with 9 different left contexts.

Recognition experiments were performed for a single speaker using three different models: context-independent (phoneme), left-context only, and a combined model. For each case, the system was alternately trained with 1, 4, 10, and 20 tokens per letter. The recognition was performed using a best-first stack search. The results [7] show that the context-dependent model performance increases significantly from 61% correct with one token per letter to 97% with 20 tokens per letter, while the context-independent model performance only varies from 79% to 93%. Thus, for small amounts of training, the context-independent model is better than the context-dependent models; for large amounts, vice versa. The combined model, with weights based on the number of tokens and the state within the model, is generally better than either model by itself, with performance ranging from 82% correct with one token to 97% with 20 tokens.

2.2.2 Continuous Phonetic Recognition

In this section, we describe experiments on continuous phonetic recognition, using the same techniques for modeling coarticulation. The analysis methods were the same as for the previous experiments with the following exceptions. The clustering and vector quantization of the speech used several different size codebooks, from 64 to 512. A simple variable-frame-rate (VFR) algorithm was used to reduce the computation somewhat. Strings of up to 3 identical vector codes were compressed to 1 observation. (This simple variable frame rate scheme was found not to affect performance.) Training sets of 5 minutes and 25 minutes were used.

In general, the models that are derived from a combination of the phoneme model and either the left or right context-dependent model resulted in significantly better performance than either the context-independent phoneme model or the left-context model alone. The system that used a combination of models dependent on left and right context simultaneously did not improve performance any further. A careful examination of the results showed that including either left context or right context produced similar answers.

The performance improves with a finer spectral resolution in the VQ codebook, as long as the training set is sufficiently large. With only five minutes of training, performance improved as the number of spectral templates increased from 64 to 256. However, for the combined model, the performance dropped when the number increased to 512 spectral templates, presumably due to insufficient training data for each pdf. As the amount of training was increased to 25 minutes, the performance improved most for those systems that used combined models and a large number of spectra. In particular, the combined phoneme+left+right context model, with 256 spectra and 25 minutes training, cut the errors in half (81% correct) relative to the context-independent (phoneme) model alone (62%).

2.2.3 Coarticulatory Effects in a Word Recognition System

In this section, we extend the coarticulation model to the problem of continuous large-vocabulary word recognition. In our phonetic recognition experiments we have observed that the improvement in performance due to using left- or right-dependent models of phonemes instead of context-independent models is smaller when the test vocabulary is different from the training vocabulary, even though the contexts in the test set had occurred frequently in the training set. We hypothesized that contexts beyond the immediate phonetic contexts are important and affect recognition results. This might explain why speech recognition systems that model whole words typically outperform those that use a phoneme model, as long as the amount of training for each word is sufficient and the effects between words are not severe. However, word-based systems cannot easily take into account word boundary effects and are not easily extensible to vocabularies of thousands of words. The problem then is to model phonemes in context to maximize recognition performance on a particular large vocabulary, especially when not all the words in the vocabulary appear often enough in the training set to allow the estimation of robust models.

To extend our model of coarticulation to the word level, we need only include a word-dependent model of the phoneme with any other models that we choose to use. We also must expand our dictionary pronunciations to permit modeling of the desired context.

Database

The vocabulary used in this study was from a 334-word electronic mail task. The task has a fairly rich structure and allows many different types of questions and commands, such as:

- Print all messages from Smith on the Dover.
- Which messages have I deleted since yesterday?
- Has Jones replied to my last message?

A total of 400 different sentences were generated covering 250 words of the vocabulary. The sentences were each recorded by three male speakers and one female speaker in sessions of 100 sentences, separated by a few days. The first three sessions were designated as training data, and the last as test material. The total duration of the training material was thus about 15 minutes for each speaker. The test material used in the experiments below included 30 of the test sentences, with a total of 187 word tokens covering 80 different words.

A dictionary of phonetic pronunciations was constructed for this 334-word vocabulary without listening to either the training or test material, but by trying to account for the most frequent phonological variations for each word. The average number of different pronunciations per word was 2. Word boundary phonological variations were *not* included. (In a separate experiment, each word was allowed only one pronunciation. The recognition accuracy was slightly higher than with multiple pronunciations. We have not fully explained this result, and are not sure whether it will carry over to very large vocabulary experiments.)

Analysis

The sentences were read directly into a close talking microphone in a natural but deliberate style in a quiet office environment. As before, some of the training data was used with a clustering algorithm to produce a representative set of Mel-Frequency Cepstral Coefficient (MFCC) vectors. However, in this case we used a k-means clustering, which was found to result in slightly better performance than the nonuniform binary clustering procedure. These experiments were performed using a codebook size of 256 MFCC templates. We used the simple VFR algorithm described above to save computation.

Training

To obtain the necessary initial estimate for the probability distribution function for each state of the phonetic HMM we use a bootstrapping technique. A separate passage (5 minutes of speech of a different vocabulary) spoken by one of the male talkers is carefully labeled, indicating the beginning frame of each phoneme. The hand-labeled speech is then quantized using the VQ codebook for each particular talker in the experiment. Normalized histograms of the observed vector-quantized spectra for each phoneme are computed from the labeled data to form an initial estimate of the pdf for the phoneme for that talker. All the pdfs for the different states in the HMM for a phoneme are set to this initial estimate. Finally, all the pdfs for the context-dependent models of a phoneme are set equal to the single, context-independent model of that phoneme. This bootstrapping technique of using a single talker's speech as an initial estimate for all talkers seems to work quite well for both male and female talkers.

We have also used a second bootstrapping technique that gives approximately the same performance without any manual labeling effort. We start from a flat initial estimate for each phoneme, and train the system using context-independent models only until convergence. Then, these models form the initial estimate for the context-dependent models, which are then trained further. This second method requires more computation, because of the need for two training sequences, but makes no assumptions about the nature of the acoustic environment, or the availability of manually labeled speech.

The 15 minutes of training data per talker is transcribed with the sequence of words spoken (no time labels and no phonetic labels). The training data is then processed with five passes of the Forward-Backward algorithm, which is normally sufficient for convergence. In the cases where context-dependent models of the phonemes are used, the training algorithm maintains separate models for each observed phonetic context. The numbers of different acoustic models found in the training set were: 50 context-independent, 500 left- or right-context dependent, and 1600 word-dependent models.

Prior to recognition, word models are precomputed for each word in the vocabulary from the appropriate phoneme-in-context models with weights depending on the number of occurrences of each model and the position within the phoneme (as used in training).

Recognition

The recognition algorithm used was a time-synchronous approximate procedure developed under the Multiple Knowledge Sources contract within the Strategic Computing Program. No grammar was used, thus making the branching factor equal to the vocabulary size (334). The recognized sequence of words was then compared automatically to the correct answer to determine the percentage of correct, deleted and inserted words. Word substitutions and deletions are tabulated as errors, while insertions are counted separately.

We present results for several different context models. As described in the previous section, the results were produced for the following set of conditions: 3 speakers, speaker-dependent, 334-word lexicon, electronic mail task, no grammar, 15 minutes of training, and 30 test utterances totaling 187 words. Table 1 gives a detailed description of the various system configurations for the different experiments.

Figure 3 shows the word recognition accuracy for each coarticulation model (identified below the graph). The left and right axes show the percentage of words correct and percent error correspondingly. This performance measure only takes into account substitution and deletion errors. Therefore, the percentage insertion errors (i.e. the number of extra words divided by the number of words spoken) is given directly above each label. For each coarticulation model, the performance is indicated for each male speaker by a filled circle. The average performance across speakers is indicated by the horizontal line. Finally, the single triangle for system PH+W indicates the recognition performance for the female talker. For this best system, the word recognition accuracy, averaged across the four speakers, was 90%.

System Name	*Word models are constructed using:*
PH	Context-indepenuent phoneme models
W	Only word-dependent phoneme models, regardless of whether training is sufficient for the word
PH+W	Linear interpolation of context-independent and word-dependent phoneme models
PH+L+R	Linear interpolation of context-independent, left-context-dependent and right-context-dependent phoneme models.
PH+L+R+W	Linear interpolation of context-independent, left-context-dependent, right-context-dependent, and word-dependent phoneme models.

Table 1: Different System Configurations for Word Recognition.

From the results given above, we make the following observations. First, the systems that model coarticulatory effects clearly result in better recognition performance. For example, system W achieves significantly better performance than system PH. Note that in this experiment, while not all vocabulary words were in the training set, all words in the 30 test sentences were observed at least once in the training. Although some words are poorly trained, the overall performance is improved. Note that for larger vocabularies, many words would not occur in training, making this system (W) inappropriate; a system that uses a subword context-dependent model will be necessary. Second, the systems that use less detailed models to smooth the highly context-dependent models result in higher accuracy and fewer insertions than those that attempt to use the context-dependent model by itself. For example, system PH+W outperforms system W. Third, the range in performance across the three speakers (17%) is large for the context-independent (PH) system. We conjecture that this is due to a difference in the degree of coarticulation present. However, the range in performance for the context-dependent systems (4-6%) is greatly reduced - a desirable attribute. We believe this behavior is due to the fact that these systems are better able to model the coarticulation present.

As a side note, we tried combining all four models (PH+L+R+W) in a single experiment, but found that performance did not improve over the PH+W system. We presume that this is due to the fact that most words in the test set were well trained.

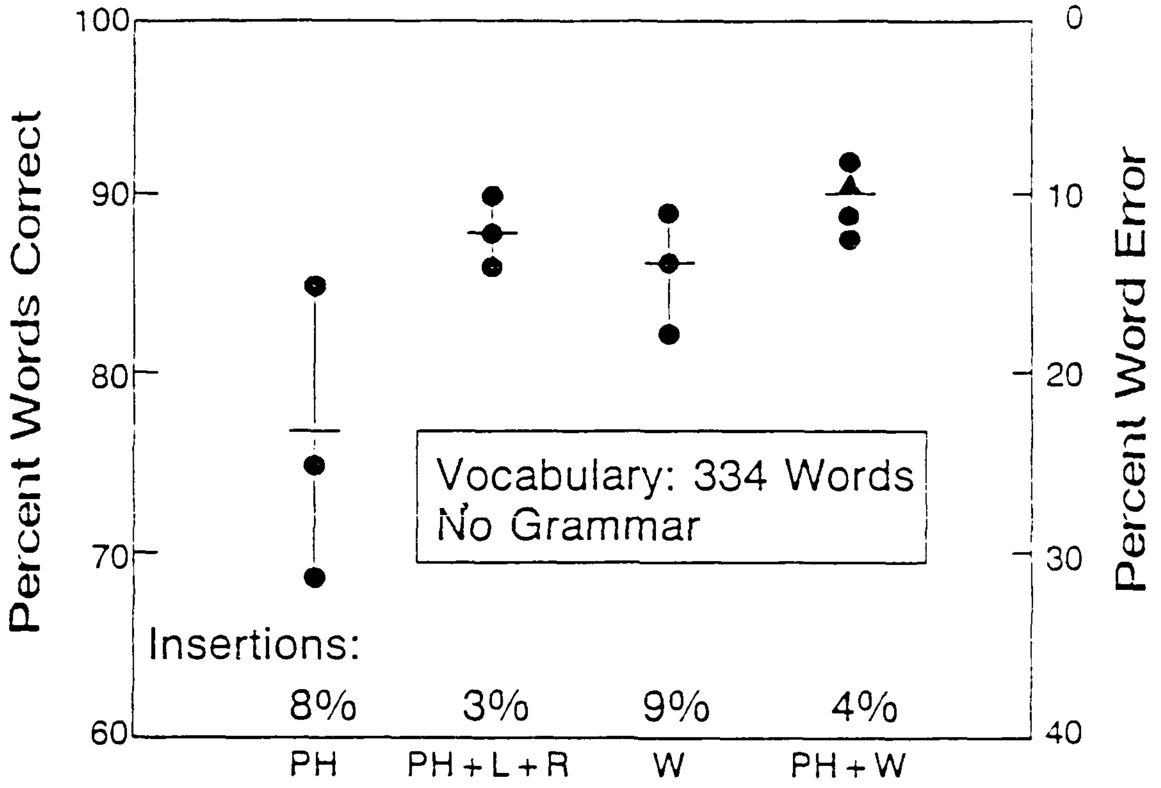

Figure 3: Word Recognition Accuracy using different coarticulation models.

2.2.4 Recognition With a Grammar

Our next goal was to show that the coarticulation model improves recognition accuracy in a complete continuous speech recognition system. We constructed a deterministic grammar for the electronic mail task using an extended context free notation. The rules were compiled into a finite-state network. The Test Set Perplexity [4] of the grammar was 31.

Table 2 compares the recognition accuracy for coarticulation models PH and PH+W. The table gives both the word accuracy (percentage of words correctly recognized) and the sentence accuracy (percentage of sentences recognized exactly correct with no insertions, substitutions, or deletions). The results are averaged across the 3 male and 1 female speakers. As seen, system PH+W, has about one fourth the word errors, and less than one third the sentence errors of system PH. The word recognition accuracy with a grammar was 98.8%, averaged over the four speakers. The sentence recognition accuracy was over 90%.

Context	*Word Accuracy*	*Sentence Accuracy*
PH	94.7%	66.4%
PH+W	98.8%	90.2%

Table 2: Recognition results with a grammar.

2.2.5 Conclusion

We have presented a formalism for modeling coarticulatory effects in a robust way. The formalism uses detailed context-dependent models of phonemes smoothed by more robust context-independent models, with weights that depend on the amount of training of each model and the location within the phoneme. Thus, the phonetic modeling in the recognition system is not tied to any particular level of context, such as the diphone or syllable. It attempts to use the information in the training data to the extent possible. Comparative results have been compiled for four different tasks: Isolated E-set recognition, continuous phonetic recognition, continuous word recognition, and continuous speech recognition using a grammar. The speaker-dependent recognition accuracies for these four problems were: E-set: 97%, phoneme recognition: 81%, continuous speech word recognition (no grammar): 90%, and continuous speech recognition with a grammar (text-set perplexity of 31): 98.8%. In all cases, the benefit of using the robust coarticulation model over the simple context-independent phonetic model was a reduction of the error rate by at least a factor of two and often more.

2.3 BYBLOS System Recognition Results

The algorithms developed under this contract were used in the BYBLOS speech recognition system developed under the Strategic Computing program. In this section we present recognition results for the BYBLOS system. In all cases, the system used the robust coarticulation model described in the previous section. We performed experiments on three different databases: the 334-word electronic mail task, a 350-word subset of the resource management task, and the 1000-word resource management task-domain database collected at Texas Instruments (TI) and at BBN. For each task, a deterministic grammar was constructed using an extended context free notation. The rules were compiled into a finite-state network. To do this, we disallowed infinite recursion. Each arc in this network represents a word and each path through the network represents a valid sentence in the language. We measured the complexity of the network by computing the Test Set Perplexity on an independent set of sentences. First, each sentence in a test set is parsed by the grammar. Then we compute the geometric mean of the number of possible words at each node of the grammar, sampled over the test set. While the perplexity of a language does not take into account the acoustic confusability of the competing words, we feel that the Test Set Perplexity measure is still a good *rough* measure of task difficulty.

The recognition algorithm used was the same time-synchronous as before, with the modification that each word-arc could only be followed by those word-arcs allowed by the grammar. While the computation for a large grammar would increase proportionally with the number of arcs in the grammar, we found that it was possible to prune most of the paths using a beam search, without any loss in performance.

Table 3 shows recognition results for several different tasks. For each task, we indicate the source of the database (how many male and female speakers, and whether recorded at BBN or at TI) and the perplexity of the grammar used. For each case, we also show the performance without a grammar (indicated by a grammar with 1 node). Finally, we give the number of nodes and arcs in the grammar.

We draw the following general conclusions from the results in the table. First, the recognition results for the case of no grammar vary between 65% and 90% correct depending on the vocabulary size and the population of speakers. These results, we feel, are unsurpassed in the literature. The recognition results with various grammars vary quite predictably as a function of the perplexity. Generally, the percent word error is approximately predicted by the formula:

$$\%error = 0.5 \times perplexity^{0.5}$$

Task	Perplexity	# Nodes	# Arcs	Speakers	% Accuracy
334 Word E-Mail	334	1	334	3M, 1F 15 min.	90.0
	31	600	4.000		98.8
350 Word Resource Management	350	1	350	2 Males from BBN	92.5
	30	7.000	30.000		99.2
1.000 Word Resource Management	1,000	1	1.000	2M(BBN)	86.2
				3M.1F(TI)	65.9
	50	1.000	1.000	BBN	98.6
				TI	89.7
	10	7.000	70.000	BBN	99.7
				TI	97.8

Table 3: Continuous speech recognition results with grammars.

or the word error rate is approximately equal to one-half the square root of the perplexity.

The accuracy for the BBN speakers appears to be a consistently higher than for the database collected at TI. While the number of speakers tested is relatively small, we believe that the difference in results between the BBN and TI speakers is largely due to the fact that the BBN speakers were more motivated and spoke more carefully than the speakers at TI.

2.4 Summary

We have described a comprehensive model for phonetic coarticulation and showed that it improves recognition accuracy for several phonetic recognition and word recognition tasks. In general, the error rate for the robust coarticulation model was less than half the error rate for the context-independent model alone. We have demonstrated that when the algorithms developed under this contract were applied to the speech recognition system developed under Strategic Computing, they resulted in high recognition accuracy on several tasks, using several different databases.

3. Stochastic Segment Modeling

Although the HMM approach has been used successfully [4, 8, 9], its recognition performance at present is not sufficiently accurate for high-perplexity continuous speech recognition. Recently, we began investigating a novel approach, called stochastic segment modeling, with the goal of improving phonetic modeling. The motivation for looking at speech on a segmental level, rather than on a frame-by-frame basis as in HMM,is that we can better capture the spectral/temporal relationship over the duration of a phoneme. Evidence of the importance of spectral correlation over the duration of a segment can be found in the success of segment-based vocoding systems [10].

A speech "segment" is a variable-length sequence of feature vectors, where the features might be, for example, cepstral coefficients. The stochastic segment model is defined on a fixed-length representation of the observed segment, which is obtained by a time-warping (or resampling) transformation. The stochastic segment model is a multivariate Gaussian density function for the resampled representation of a segment. The recognition algorithm chooses the phoneme sequence that maximizes a match score on the resampled segments. The training algorithm iterates between two steps: first, the maximum probability phonetic segmentation of the input speech is obtained, then maximum likelihood density estimates of the segment models are derived.

This section is organized as follows. First, we introduce the segment model followed by a description of the segment-based recognition algorithm, then the training algorithm. Finally, we present experimental results for phoneme and word recognition, comparing the results to HMM recognition results for the same tasks.

3.1 Stochastic Segment Model

In this section, we define the stochastic segment model for an observed sequence of speech frames $X = [x_1\ x_2 \ldots x_L]$, where x_i is a k-dimensional feature vector. We can think of this observation as a variable-length realization of an underlying fixed-length spectral trajectory $Y = [y_1\ y_2 \ldots y_m]$ where the duration of X is variable due to variation in speaking rate. Given X, we define the fixed-length representation $Y = XT_L$ where the L x m matrix T_L, called the resampling transformation, represents a time-warping. The segment Y, called a *resampled* segment, is an m-long sequence of k-dimensional vectors (or a k x m matrix). The stochastic segment model for each phoneme α is based on the resampled segment Y and is a conditional probability density function $p(Y|\alpha)$. The density $p(Y|\alpha)$ is assumed to be multivariate Gaussian which is a km-dimensional model for the entire fixed-length segment Y.

3.1.1 Resampling Transformations

The resampling transformation T_L is an L x m matrix used to transform an L-length observed segment X into an m-length resampled segment Y. We considered several different variable- to fixed-length transformations, concentrating on transformations which had previously been evaluated in the segment vocoder [10]. The best recognition results are obtained using linear time sampling without interpolation. Linear time sampling involves choosing m uniformly spaced times at which to sample the segment trajectory. Sampling without interpolation refers to choosing the nearest observation in time to the sample point, rather than interpolating to find a value at the sample point.

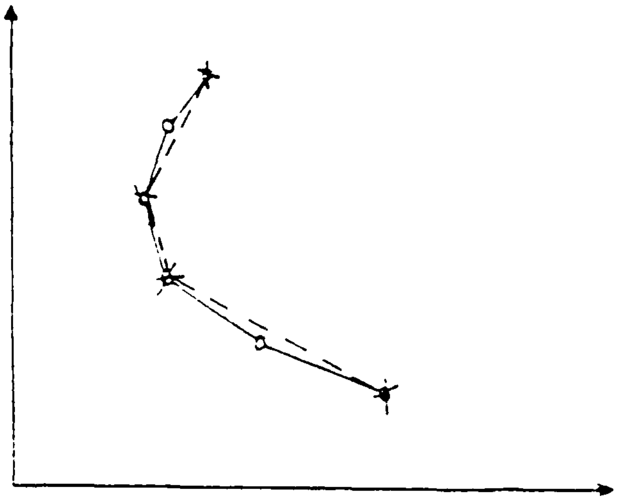

Figure 4: Observed input segment (o) and corresponding resampled segment (x). The two axes correspond to two cepstral coefficients.

Figure 4 shows an input segment with duration six in two-dimensional space (denoted by o) and the corresponding resampled Y (with m = 4) using linear time warping without interpolation (denoted by x). The resampling transformation in this case is:

$$T = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

3.1.2 Probabilistic Model

As already mentioned, the segment model is a multivariate Gaussian based on the resampled segment Y, $p(Y|\alpha)$. Recall that resampled segments are km-dimensional, where k is the number of spectral features per sample and m is the number of samples. In this work, typically $k=14$ and $m=10$. Consequently, the segment model has 140 dimensions. Because of insufficient training, we cannot estimate the full phoneme-dependent covariance matrix, so we must make some simplifying assumptions about the structure of the problem. For the experiments reported here, we assume that the m samples of the resampled segment are independent of each other, which gives a block diagonal covariance structure for Y, where each block in the segment covariance matrix corresponds to the $k\ x\ k$ covariance of a sample. The log of the conditional probability of a segment Y given phoneme α can then be expressed as

$$ln[p(Y|\alpha)]=\sum_{j=1}^{m} ln[p_j(y_j|\alpha)], \qquad (4)$$

where $p_j(y_j|\alpha)$ is a k-dimensional multivariate Gaussian model for the j-th sample in the segment. The block-diagonal structure saves a factor of m in storage and a factor of m^2 in computation. The disadvantage of this approach is that the assumption of independence is not valid, particularly if resampling does not use interpolation where adjacent samples may be identical. In the future, with more training data, we hope to relax this assumption. It is likely that more detailed probabilistic models, such as Gaussian mixture models [11] and context-dependent (conditional) models [8, 9], will yield better recognition results than the simple Gaussian model. However, due to larger training requirements we did not pursue these models in this work.

3.1.3 Properties of the Segment Model

There are several aspects of the stochastic segment model which are useful properties for a speech recognition system. First, the transformation T_L, which maps the variable-length observation to a fixed-length segment, can be designed to constrain the temporal structure of a phoneme model so that all portions of the model are used in the recognition. We conjecture that the fixed transformation will provide a better model of phoneme temporal/spectral structure than either HMM or DTW. Second, the segment model is a joint representation of the phoneme, so the model can capture correlation structure on a segmental level. In HMM, frames are assumed independent given the state sequence. In the segment model, no assumptions of independence are *necessary*, though the model of Y given by Equation 4 is based on the assumption of sample independence because of limited training data in this study. The model is potentially more general than the special case of (4). Lastly, by using a segment model we can compute segment level features for phoneme recognition. In other words, the segment model provides a good structure for incorporating acoustic-phonetic features in a statistical (rather than rule-based) recognition system. For example, one might want to measure and incorporate formant frequency or energy differences over a segment.

3.2 Recognition Algorithm

In this section, we describe the recognition algorithm. First, we consider the case when the input is phonetically hand-segmented. Then, we generalize to automatic recognition, that is, joint segmentation and recognition of continuous speech.

When the segmentation of the input is known, we consider a single segment X independently of neighboring segments. The input segment X is resampled as segment Y. The recognition algorithm is then to find the phoneme $\hat{\alpha}$ that maximizes $p(Y|\alpha)$:

$$\hat{\alpha}=arg \max_{\alpha} ln[p(Y|\alpha)p(\alpha)] \tag{5}$$

where $\ln[p(Y|\alpha)]$ is given by Equation 4. This decision rule is equivalent to a maximum a-posteriori rule.

In an automatic recognition system, it is necessary to find the segmentation as well as to recognize the phonemes. In this case, we hypothesize all possible segmentations of the input, and for each hypothesized segmentation $\underline{s}$ of the input into n segments we choose the sequence of phonemes $\hat{\underline{\alpha}}$ that maximizes:

$$J(\underline{s})=\sum_{i=1}^{n} L(i) \tag{6}$$

$$ln[p(Y_i|\hat{\alpha}_i)p(\hat{\alpha}_i]+nC$$

where L(i) is the duration of the i-th segment, Y_i is the resampled segment corresponding to the i-th segment in $\underline{s}$, and $\hat{\alpha}_i$ is the phoneme that maximizes $p(Y_i|\alpha)p(\alpha)$. The cost C is adjusted to control the segment rate. An efficient solution to joint segmentation and recognition is implemented using a dynamic programming algorithm. The unoptimized algorithm for finding the best sequence of segment models to the input is described roughly by the pseudocode given below

```
for end_time = 4 to Number_of_frames

/* score from frame 0 to end_time
{
  best_score[end_time] = 0
/* consider input segments 4 to 50 frames ending at end_time */
  for begin_time = end_time - 4 to end_time - 50 by -1

  /* find best segment model between begin_time and end_time */
  {
    time_warp_input(begin_time,end_time, time_warped_input)
    max_prob_seg = 0
    for iseg = 1 to Number_of_segment_models
      log_prob = log p (time_warped_input | segment[iseg])
      if (log_prob > max_prob_seg) then
      { max_prob_seg = log_prob
        best_seg = iseg
       }

    /* score of 1 to begin_time + begin_time to end_time */
    best_begin_end_score = best_score[begin_time]
                 + max_prob_seg * (end_time - begin_time)
    if (best_begin_end_score > best_score[end_time]) then
    {  best_score[end_time] = best_begin_end_score
       best_start_time[end_time] = begin_time
       best_seg_ending[end_time] = best_seg
     }
   }
}
```

Note that for joint segmentation and recognition, it is necessary to weight the segment probability by the duration of the segment, so that longer segments contribute proportionately higher scores to the match score J(.) of the whole sequence.

3.3 Training Algorithm

In this section, we present the training algorithm for estimating the segment models from continuous speech. We assume that the phonetic transcription of the training data is known and that we have an initial Gaussian model $p_0(Y|\alpha)$ for all phonemes. (Phonetic transcriptions can be generated automatically from the word sequence that corresponds to the speech by using a word pronunciation dictionary.) We assume that the phonetic sequence $\underline{\alpha}$ has length n. The algorithm comprises two steps: automatic segmentation and parameter estimation. The algorithm maximizes the log likelihood of the optimal segmentation for the phonetic transcription, where the log likelihood of a segmentation $\underline{s}$ is given by:

$$l(\underline{s})=\sum_{i=1}^{N} ln[p(Y_i|\alpha_i)p(\alpha_i)] \qquad (7)$$

where $\mathbf{Y}_i$ is the resampled segment that corresponds to the i-th segment in the segmentation $\underline{s}$ and α_i is the i-th phoneme in the sequence $\underline{\alpha}$. With t = 0, the iterative algorithm is given by:

1. Find the segmentation $\underline{s}_t$ of the training data that maximizes $l(\underline{s}_t)$ for the given transcription and the current probability densities $\{p_t(Y|\alpha)\}$.
2. Find the maximum likelihood estimate for the densities $\{p_{t+1}(Y|\alpha)\}$ of all phonemes, using the segmentation $\underline{s}_t$.
3. t <- t + 1 and go to Step 1

Both steps of the algorithm are guaranteed to increase $l(\underline{s}_t)$ with t. If there are at least two *different* observations of every phoneme, then the probability of the sequence is bounded. Hence, the iterative training algorithm converges to a local optimum. Step 1 is implemented as a dynamic programming search whose complexity is linear with the number of phonetic models N. Step 2 is the usual sample mean and sample covariance maximum likelihood estimates for Gaussian densities.

3.4 Experimental Results

In this section we will present results for a phoneme recognition task, as well as word recognition results for a segment-based recognition system and an HMM-based system. All experiments use m = 10 samples per segment and k = 14 mel-frequency cepstral coefficients per sample. These values are based on work in segment quantization [12], and limited experimentation confirmed that these values represent a reasonable compromise between complexity and performance. Speech is sampled at 20 kHz, and analyzed every 10 ms with a 20 ms Hamming window.

3.4.1 Phoneme Recognition

The database used for phoneme recognition is approximately five minutes of continuous speech from a single speaker. The test set contains 270 phonemes. Both the test set and the training set are hand-labelled and segmented, using a 61 symbol phonetic alphabet. In counting errors, an 'AX' (schwa) recognized as 'IX' (fronted schwa) is considered acceptably correct, as is an 'URT' (unreleased T) recognized as a 'T'. All recognition rates presented represent "acceptably correct" recognition rates. The acceptable recognition rate is typically 6% to 8% higher than the strictly correct recognition rate.

Phoneme recognition results for three different cases are given in Table 4. The results illustrate a small degradation in performance due to moving from recognition based on manually segmented data to automatic recognition. Using automatic training does not degrade performance any further.

Training Segmentation	Test Segmentation	% Recognition	% Insertion
Manual	Manual	78.5	0.0
Manual	Automatic	74.4	10.0
Automatic	Automatic	73.7	7.8

Table 4: Recognition results using manually segmented speech and automatically segmented speech.

We also experimented with using an additional segmental feature to the cepstral parameters: sample duration which requires knowledge of the hypothesized duration of the segment. Using joint segmentation and recognition with hand-segmented training data, performance improved from 74.4% to 75.9% as a result of using the duration feature.

For reference, a discrete hidden Markov model with 3 states/phoneme and using a codebook with 256 entries has 62% phonetic recognition rate with 12% insertions. The HMM recognition performance on this database is higher when phoneme models are conditioned on left context, 75% correct with 12% insertions [8]. In the latter case, 600 left-context phonetic models are used in the HMM system while 61 phonetic models are used in the stochastic segment model.

3.4.2 Word Recognition

The segment-based word recognition system consists of a dictionary of phoneme pronunciation networks and a collection of segment phoneme models. A word model is built by concatenating phoneme models according to the pronunciation network. The recognition algorithm is simply a dynamic programming search (Viterbi decoding) of all possible word sequences. For the results in this paper, we assume that words are independent and equally probable; there is no grammar (statistical or deterministic) associated with recognition. Within each word, we find the best phoneme segmentation for that word, where the phoneme sequence is constrained by the word pronunciation network.

For continuous speech word recognition, we used a 350 word vocabulary, speaker-dependent database based on an electronic mail task. We present results for three different male speakers. Fifteen minutes of speech was used for training the 61 phoneme models for each speaker, from which the word models were then built. An additional 30 sentences (187 words) are used for recognition. Analysis parameters are the same as for the previous database. Again, "acceptable" error rates are reported here, where in this case, homophones such as "two" and "to" are considered acceptable errors. Since we do not use a grammar, homophones are indistinguishable.

The initial segment models are obtained on training from segmentations given by a discrete hidden Markov model recognition system. The results after one pass of training of the segment model for the three speakers are summarized in Table 5. The HMM recognition results are also given for comparison. For the HMM results, five passes of the forward-backward training algorithm are performed. The segment phoneme system outperforms the phoneme-based HMM system, reducing the error rate by one third (including insertions). However, the segment phoneme system does not quite match the HMM context model system. This suggests that context-dependent segment models might be useful. Note that in the earlier phoneme results, the segment system matched the performance of HMM models conditioned on left context only. Here we give results for HMM models conditioned on both left and right context. The HMM system with context models conditioned on both left and right context uses 2000 models, or thirty times the number used by the segment system.

Speaker	Segment- PH	HMM- PH	HMM-OP PH-LE-RI
RS	87/5.3	85/10.2	90/1.1
FK	83/2.1	75/ 5.4	88/2.7
AW	78/3.7	68/ 7.5	86/3.7
Average	83/3.7	76/ 7.7	88/2.5

Table 5: Word recognition/insertion rates for three speakers for the segment phoneme system and for two HMM systems: phoneme models and phoneme models conditioned on the left and right context.

3.5 Conclusion

To summarize, we feel that the segment model offers the potential for large improvements in speaker-dependent acoustic modeling of phonemes in continuous speech. Our initial results demonstrate the potential of the approach. Of course, a practical system requires automatic training and recognition, which we demonstrated to perform close to the hand-segmented case at the cost of a few insertions. For comparison, the automatic segment system reduces the word error rate by one third over an HMM system on a 350-word continuous speech recognition task.

4. Rapid Speaker Adaptation

To achieve the high recognition accuracy presented in previous sections, each speaker read 300 training sentences or about 15 minutes of training speech. Some speech recognition applications have a need for a new speaker to begin using the system with reasonable accuracy without investing a long time to train the system on their voice. However, as we will see below, the speaker-dependent performance degrades dramatically when the amount of training speech is reduced using the standard training procedure. A different training procedure is needed if we are to have robust and rapid speaker adaptation with only a few training sentences. The purpose of speaker adaptation is to yield acceptable recognition performance even for speakers who have not provided enough speech to train the HMMs. Since our goal is to achieve the highest performance possible, we focus on developing speaker adaptation procedures which operate on a set of known sentences from a new speaker (supervised training). These sentences (adaptation speech) are processed before the new speaker begins to use the system.

The approach that we have taken in our work is to normalize well-trained models from a "prototype" speaker to model the speech of the new speaker. The adaptation requires only a few sentences (referred to as "adaptation speech") from the new speaker.

Below we define a probabilistic spectral mapping from one speaker to another. Then, we present two different methods for estimating the spectral mapping between speakers. With each method, we present experimental results.

4.1 Probabilistic Mapping

In this section, we describe our basic approach for solving the speaker adaptation problem. We concentrate on methods which transform well-trained HMMs from a prototype speaker to model a new speaker using a probability transformation matrix.

We start with a set of well-trained speaker-dependent phonetic HMM models derived from a large sample of speech from a prototype speaker. We assume we are given a small sample of known speech from a new speaker (input speaker). The essential idea is to modify the prototype HMM parameters using a constrained transformation to model the input speech of the new speaker.

Here we present the basis for the probabilistic transformation and show it to be equivalent to an expanded HMM model for each state of the original HMM. The transformation is generalized to be partially dependent on the particular phoneme.

Discrete Hidden Markov Models

For each state of a discrete HMM, we have a discrete probability density function (pdf) defined over a fixed set, N, of spectral templates. For example, in the BYBLOS system we typically use a vector quantization (VQ) codebook of size N=256 [6]. The index of the closest template is referred to below as the "quantized spectrum". We can view the discrete pdf for each state s as a probability row vector

$$\underline{p}(s) = [p(k_1|s),\ p(k_2|s),\ \ldots,\ p(k_N|s)], \tag{8}$$

where $p(k_i|s)$ is the probability of spectral template k_i at state s of the HMM model.

Mapping From Prototype to New Speaker

If we define a quantized spectrum for the prototype speaker as k_i, $1 \leq i \leq N$, where i is the index of the spectral template and a quantized spectrum for the new speaker as k'_j, $1 \leq j \leq N$, then we denote the probability that the new speaker will produce quantized spectrum k'_j, given that the prototype speaker produced spectrum k_i, as $p(k'_j|k_i)$ for all i, j.

We can rewrite the probability for spectrum k'_j given a particular state s of the HMM as

$$p(k'_j|s) = \sum_{i=1}^{N} p(k_i|s)\ p(k'_j|k_i,s) \tag{9}$$

If we assume that the probability of k' given k is independent of s, then

$$p(k'_j|s) = \sum_{i=1}^{N} p(k_i|s)\ p(k'_j|k_i) \tag{10}$$

The set of probabilities $p(k'_j|k_i)$ for all i and j form an $N \times N$ matrix T that can be interpreted as a probabilistic transformation from one speaker's spectral space to another's. We can then compute the discrete pdf $\underline{p}'(s)$ at state s for the new speaker as the product of the row vector $\underline{p}(s)$ and the matrix T.

$$\underline{p}'(s) = \underline{p}(s)\ \mathrm{T}; \qquad \mathrm{T}_{ij} = p(k'_j|k_i) \tag{11}$$

Expanded HMM Formulation

The probabilistic transformation can also be described in terms of an expanded HMM model for the state. Figure 5a shows a single state of the HMM for a new speaker. It contains a single discrete probability vector, $\underline{p}'(s)$. Figure 5b shows an expanded model in which the single state is replace by N parallel paths. The transition probability for path i is $p(k_i|s)$, the probability of the quantized spectrum k_i, given the same state s for the prototype speaker. The discrete pdf on that path is $\underline{p}(k'|k_i)$, which corresponds to row i of the transformation matrix.

Careful inspection of the figure will reveal that the probability of any new-speaker spectrum k'_j for the expanded HMM shown is a summation of the jth probability over all N paths, as given in (10). Therefore, Figure 5a represents the left side of equation (11), while

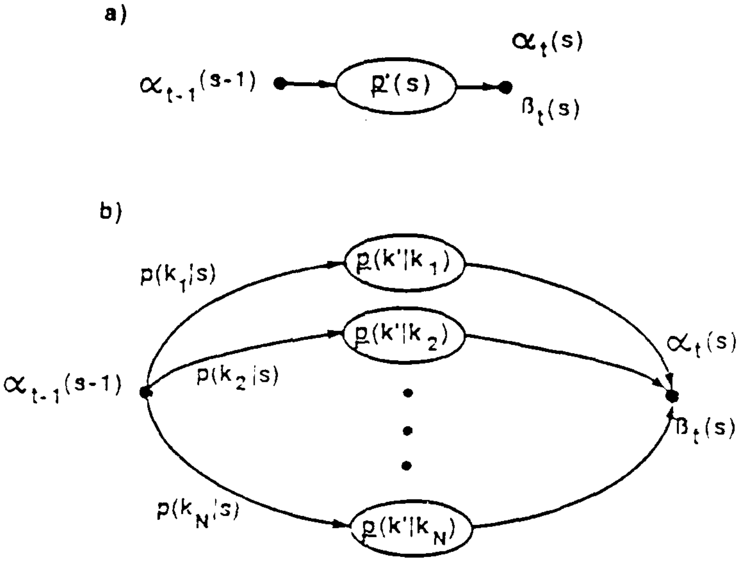

Figure 5: Expanded HMM. a) single state of the HMM; b) expanded model separating prototype pdf and transformation matrix.

figure 5b represents the right side. Once the matrix has been determined, we can replace the expanded HMM by the single pdf resulting from the vector-matrix multiplication in (11).

Phoneme-Dependent Transformation

The independence assumption in (10) above assumes that a single probabilistic spectral mapping will transform the speech of one speaker to that of another. However, we know that some of the differences between speakers cannot be modeled this simply. We can define a phoneme-dependent mapping:

$$p(k'_j|s) = \sum_{i=1}^{N} p(k_i|s) \; p(k'_j|k_i,\phi(s)) \qquad (12)$$

where $\phi(s)$ specifies the equivalence class of states in models that represent the same phoneme as s. Since the amount of training speech from the new speaker will be small, we could not hope to have enough samples of each phoneme to estimate a reliable mapping for all phonemes. Therefore, we interpolate the *phoneme-dependent* transformation matrix with the *phoneme-independent* transformation matrix. The weight for the combination depends on the number of observed frames of the particular phoneme. Thus for those phonemes that occur several times in the adaptation speech, the transformation will depend mostly on that particular phoneme.

4.2 Estimation of Transformation Matrix

We have shown above that the transformation matrix can be explained as an expanded HMM for each state of the model for the new speaker. Therefore, it would seem reasonable to use the forward-backward algorithm to estimate the transformation matrix while keeping the prototype pdf fixed. This algorithm, and the corresponding experimental results are given below.

4.2.1 Method

The algorithm begins with a VQ codebook and well-trained *context-dependent* and *context-independent* pdfs derived from a prototype speaker. A small number of sentences are read by the new speaker. The new (adaptation) speech is quantized using the prototype speaker's VQ codebook. (This step may be a source of reduced performance, and will be discussed further below.) Then, we use a modification of the standard forward-backward algorithm to estimate the *phoneme-dependent* and *phoneme-independent* transformation matrices.

To save computation and storage we use $\underline{p}'(s)$, the compact HMM in Figure 5a, to compute the partial (α and β) terms in the forward-backward algorithm. The forward-backward "counts" are added to a separate count matrix. (Two methods for computing the counts are defined at the end of this subsection.) Since we have no *a priori* transformation matrix, we must provide an initial estimate. To minimize computation we use an identity matrix for the first transformation (that is, we just use the prototype pdf as is). However, when we compute the counts in the first pass, the transformation matrix is a constant value of $1/N$. After the first pass, the same matrix is used both for forward-backward partial terms and for computing the counts. At the end of each pass through the adaptation data, each row of the count matrix, which corresponds to $\underline{p}(k'|k_i)$ (the transformation given one prototype spectrum k_i), is rescaled so it sums to 1. This normalized count matrix then becomes the new probabilistic transformation matrix. After the final pass we transform all the prototype models using (11).

Computing Counts - Method 1:

For each alignment of a state with an observed quantized spectrum, $k'(t)=k'_j$, the prototype pdf vector $\underline{p}(s)$ is multiplied by column j of the transformation matrix $p(k'_j|k_i)$, $1 \leq i \leq N$. This vector product is multiplied by the constants $\alpha_{t-1}(s-1)$ and $\beta_t(s)$ (shown in Figure 5b) and then accumulated in column j of the count matrix. $\alpha_{t-1}(s-1)$ is the probability of the observed spectra from frames 1 through $t-1$ given the models up to but not including state s. $\beta_t(s)$ is the probability of the observed spectra from the end of the sentence back to time $t+1$ given the models after state s. This method corresponds to the standard (maximum likelihood) forward-backward algorithm for the HMM shown in Figure 5b.

Computing Counts - Method 2:

Method 2 is similar to Method 1, with the exception that the prototype pdf vector is multiplied by the constants $\alpha_t(s)$ and $\beta_t(s)$ (shown in Figure 5b) and then added to the corresponding column of the count matrix. That is, the counts are computed as the probability of being in state s at time t, times the prototype pdf. We found that only one pass of the algorithm is necessary for Method 2, making it preferable in terms of computation. We also found that this method results in slightly better performance than Method 1. Therefore all results quoted below are for Method 2.

4.2.2 Experiments

Database

We have performed experiments on a 350-word subset of a naval database retrieval task (FCCBMP). The task has a fairly rich structure and allows many different types of questions and commands. The prototype speaker recorded 400 sentences in 4 sessions of 100 sentences each, separated by a few days. The first three sessions were designated as training data, and the last as test material. At an average of 3 seconds per sentence, the total duration of the training material was thus about 15 minutes for the prototype speaker.

Each of 6 new speakers then recorded a subset of the training sentences and, in a separate session, the 100 test sentences. The 6 speakers included one female, one non-native speaker, one experienced speaker, and three inexperienced speakers.

We constructed a dictionary of phonetic pronunciations for the vocabulary without listening to either the training or test material. With very few exceptions, only one pronunciation was chosen for each word.

The sentences were read directly into a close-talking microphone in a natural but deliberate style in a quiet office environment. The speech was lowpass filtered at 10 kHz and sampled at 20 kHz. Fourteen Mel-frequency cepstral coefficients (MFCC) were computed every 10 ms on a 20 ms analysis window. One half of the training speech of the prototype speaker was used to derive a speaker-dependent VQ codebook. Then all the recorded speech for all speakers was quantized using this codebook.

Training

The 15 minutes of speech from the prototype speaker was used, together with the phonetic dictionary, to estimate *context-dependent* and *context-independent* phonetic models. The speech models for the new test speakers were computed in two ways: Speaker-Dependent training and Speaker adaptation. In addition to these two models for the new speaker, we also performed control experiments using the prototype speaker's models without any change. These unaltered models are designated as "Cross-Speaker" models. Prior to recognition, the phonetic models were combined and concatenated into word models to facilitate the word recognition process.

Recognition

We used the time-synchronous search procedure described in [8] to find the most likely sequence of words for each test sentence. Recognition experiments were performed both with and without a grammar. When no grammar was used, the effective branching factor was equal to the vocabulary size (350). The grammar used had a Maximum Perplexity [13] of 30 and an estimated Perplexity [4] of 20 (measured on a test set). The recognized sequence of words was then compared automatically to the correct answer to determine the percentage of errors of each type: substitutions, deletions, and insertions.

We use an error measure that reflects all three types of errors in a single number. The percent error is given by

$$\%error = 100 \frac{\text{substitutions} + \text{deletions} + \text{insertions}}{\text{total words} + \text{insertions}} \tag{13}$$

The word accuracy is then defined as 100 - $\%error$. Note that this definition is different from the percent correct words.

4.2.3 Results

Figure 6 below shows the recognition error as a function of the amount of training speech (on a log scale) for both training conditions. For reference, the results using the Cross-Speaker models are also shown. Some of the conditions that did not seem to warrant extensive testing (e.g., 15 second speaker-dependent training) were evaluated using a subset of the speakers. More critical results (e.g., 15 second speaker adaptation) were evaluated using all 6 speakers.

The recognition error varied less with the duration of speech for speaker adaptation than for speaker-dependent training - particularly when a grammar was used. The error rate with 15 seconds of adaptation speech was about the same as achieved by the speaker-dependent training method with 6 to 10 minutes of training speech. In particular, when a grammar was used, the word recognition error with only 15 seconds of adaptation speech from each speaker was 4% (97% correct words with 1% insertions.)

Detail vs Robustness

We can see from the results with and without a grammar that the speaker transformation seems to be much more successful when a grammar is used. That is, the error decreased by a

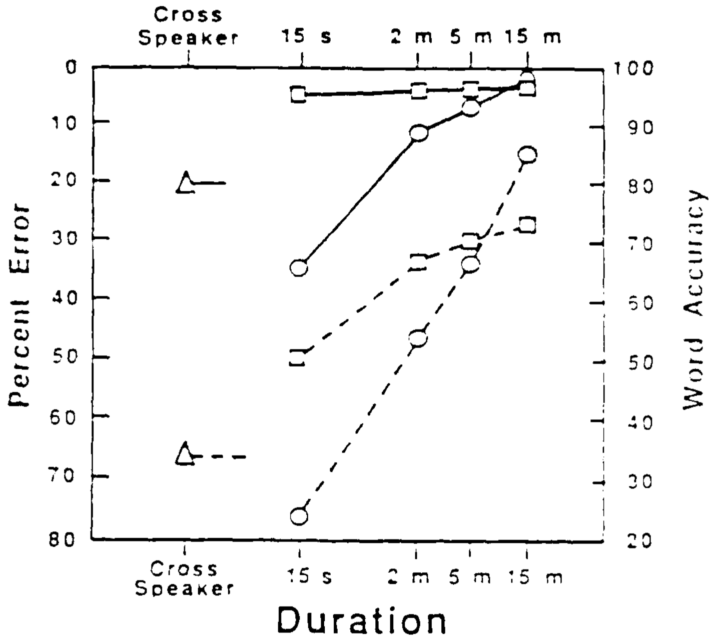

Figure 6: Speaker-Dependent Training vs adaptation.
Speaker-Dependent Training (○); Speaker adaptation (□); Cross-Speaker Results (△).
The solid line indicates accuracy with a grammar; the dashed line indicates no grammar.

bigger factor (from speaker-dependent training to the adaptation algorithm) when a grammar was used than when no grammar was used.

When no grammar is used in speech recognition it is important that the models be sharply tuned to make fine distinctions. Occassional errors will result from a finely tuned model that was inadequately trained. In contrast, we assume that when a grammar is used the number of words allowed at each point is small relative to the vocabulary size. In this case it is less likely that fine phonetic distinctions will be necessary. To get very high performance, it becomes more important that the correct word *never* get a very low score.

We have observed that the pdfs resulting from the speaker adaptation procedure are typically broader than those resulting from speaker-dependent training. We surmise that this effect, combined with the appropriate spectral mapping between the speakers, accounts for the large improvement in accuracy when a grammar is used.

Source of Errors

We performed a series of experiments on one speaker in an effort to determine whether the major source of errors is the duration of adaptation of speech, the adaptation procedure itself, or the fact that the VQ codebook of the prototype speaker is used for the new speaker. We present the recognition results (using no grammar) in Table 6 below.

Condition	*% error*
15 min spkr-dependent training	16%
Prototype VQ codebook	24%
15 min adaptation	27%
5 min adaptation	30%
2 min adaptation	33%

Table 6: Source of Recognition Errors.
Each line changes one experimental condition.

As we see in the table, the largest increase in word error is the result of using a VQ codebook that was not designed for the new speaker. Our next step, therefore, will be to derive a codebook for the new speaker from a combination of the new speech and the prototype speaker's codebook. This expanded codebook will form the basis for the normalized pdf models.

Results with Larger Grammars

More recently, we have performed limited experiments with a grammar that has higher perplexity than the grammar used in the experiments described above. The high-perplexity grammar was based on the 1000-word Resource Management task. Starting with a low-perplexity Sentence Pattern Grammar, we allowed all word pairs that could occur resulting in what we call the First Order Grammar, with a perplexity of 50. This grammar was used in several speaker-dependent recognition experiments. The recognition error ranged from 7% (for TI speakers) to 3% (for a BBN speaker other than the prototype speaker). Limited experiments were run on this grammar using phonetic models adapted from a prototype speaker to another. The recognition error went up to 35-40% for the TI speakers and to 18% for the BBN speaker. This represents a 5-fold increase in the error rate from the speaker-dependent to the speaker-adapted models. This is in contrast to the factor of 2 that we observed with the more restricted grammars. This shows that this speaker adaptation technique is not preserving the ability to make the fine phonetic distinctions necessary for more complex grammars. Below, we present another method for estimating the speaker transformation matrix that results in improved peromance for grammars with high perplexity.

4.3 Estimation of Matrix using Text-Dependent Alignment

As described above, one of the sources of degradation of the algorithm is due to the fact that we quantize the new speaker's speech using the prototype speaker's VQ codebook. This causes increased spectral quantization error of the new speaker's spectral parameters. In addition, many spectra from the prototype speaker's codebook are not typically observed in the adaptation speech from the input speaker, thus causing those columns of the transformation matrix to be empty and the estimated probability of those spectra to be zero in all the pdfs.

In this section we describe an improved procedure for estimating the pdf transformation matrix, **T**. First we present the method. Then we describe a set of experiments comparing several different methods for estimating the transformation matrix.

4.3.1 Method

Making speaker-dependent codebooks from limited speech

One solution to the VQ error mentioned above is to make a speaker-dependent codebook from the adaptation speech itself. Experiments show that a codebook made from 10 adaptation sentences is able to cover the feature space of future input speech and quantize future input speech with small distortion. On an independent test set, the VQ error is much less (equivalent to using 2 bits more) than that using the prototype speaker's codebook. The problem of empty columns in the matrix is eliminated, since the quantized adaptation speech uses all of the bins in the codebooks.

We also considered an algorithm of Shikano *et al.* [14] for adapting the prototype speaker's codebook to the new speaker. The use of this algorithm both as a method for determining the codebook for the new speaker and in combination with the our probabilistic spectral mapping algorithm will be discussed further below.

Our next problem is to find a procedure to compute a reliable spectral mapping between speech samples quantized by two independent codebooks.

Computing a mapping between spectra

Instead of aligning quantized spectra to prototype models, here we align the adaptation speech with a set of the same sentences spoken by the prototype speaker. That is, the alignment is performed text-dependently. We align the cepstral coefficients of the matched utterances directly with a dynamic time warping (DTW) algorithm using Euclidean distance.

Next we quantize the adaptation speech from both the input and prototype speakers using their respective speaker-dependent codebooks, and obtain a correspondence between sequences of aligned quantized spectra. To estimate $p(k'_j|k_i)$ we count the co-occurrences of the actual quantized spectra for each of the frames in the adaptation sentences. The result is a co-occurrence matrix $\mathbf{N}$, where each element $\mathbf{N}_{ij}$ is the number of co-occurrences of prototype spectrum k'_j and k_i. Then we normalize the rows of $\mathbf{N}$ to form the probability matrix $\mathbf{T}$.

Improving the estimate of the matrix

As described above, this new algorithm uses actual observations of spectra to estimate the pdf transformation matrix. An obvious way to improve the estimate of the transformation matrix is to use more adaptation speech. However, for rapid speaker adaptation, we want to minimize the adaptation material required. Another way to improve the reliability of the transformation matrix estimate without increasing the adaptation speech is to use repetitions of the adaptation sentences from the prototype speaker. In other words, we align each sentence in the adaptation speech repeatedly against several repetitions of that sentence by the prototype speaker.

Using more repetitions of the prototype speech does not increase the phonetic variety in the adaptation speech, but it does make the spectral mapping more reliable by enlarging the sample space over which the probabilities in the transformation matrix T are estimated. Experimental results below show that this procedure does improve the performance significantly.

4.3.2 Experiments

The Prototype Speaker

In all the adaptation experiments shown below, we use as a prototype the well-trained HMMs of a single speaker RS. RS is a careful male speaker with a New York dialect. RS recorded 600 sentences at BBN in a normal office environment. The 600 utterances constituted about 30 minutes of speech which was used to estimate the HMM parameters for the prototype models.

The test speaker database

A 1000-word database of continuous speech has been designed and recorded within the DARPA Strategic Computing Speech Program [15]. This data consists of sentences which are appropriate in a naval resource management task. A large number of speakers were recorded in a sound isolated recording booth at Texas Instruments (TI). We used 4 speakers from the speaker-dependent portion of this database to test 6 different adaptation procedures. We then used 8 speakers recorded at TI and 3 additional speakers recorded at BBN to compare the performance of the new adaptation method with speaker-dependent training.

Adaptation speech

In the adaptation experiments, we use adaptation speech of duration 30 seconds or 2 minutes. For the 30-second adaptation, we used 10 phonetically balanced sentences recorded during one of the speaker-dependent training sessions. For 2 minutes adaptation we added 30 more sentences taken from the training data.

Processing of the speech

Both the input and prototype speech were lowpass filtered at 10 kHz and sampled at 20 kHz. 14 mel-frequency cepstral coefficients (MFCC) were computed every 10 ms on a 20 ms analysis window.

Recognition

All the recognition experiments used a word-pair grammar of perplexity 60. This grammar allows all two-word sequences which occur in the task domain definition [15, 16]. The recognized sequence of words was compared automatically to the correct answer to determine the percentage of word errors of each type: substitutions, *deletions*, and *insertions*. We use an error measure that reflects all three types of errors in a single number. The percent error is given by

$$\%\ \textit{word-error} \ = \ 100 \times \frac{\textit{substitutions+insertions+deletions}}{\textit{total-number-of-input-words}}$$

Note that it is possible for this error measure to exceed 100%.

4.3.3 Results

We used 4 speakers from the TI database to compare 6 different adaptation procedures. All the test speakers have different sentence text for their test sets. Figure 7 shows the comparison of average word error rate using different adaptation algorithms as well as speaker-dependent training. Below we discuss the performance of each algorithm column by column from left to right in Figure 7.

Text-Independent Mapping (previous algorithm)

In the leftmost column, we show the performance for our previous algorithm using 30 seconds of adaptation speech. The word error rate is 52%, which is far from acceptable. The word error rate is about 6 times that of the speaker-dependent performance (9%) shown in the rightmost column.

Codebook adaptation

An algorithm that adapts the codebook of the prototype speaker to a new speaker has been suggested by Shikano *et al.* [14]. The purpose of experiments using codebook adaptation is to

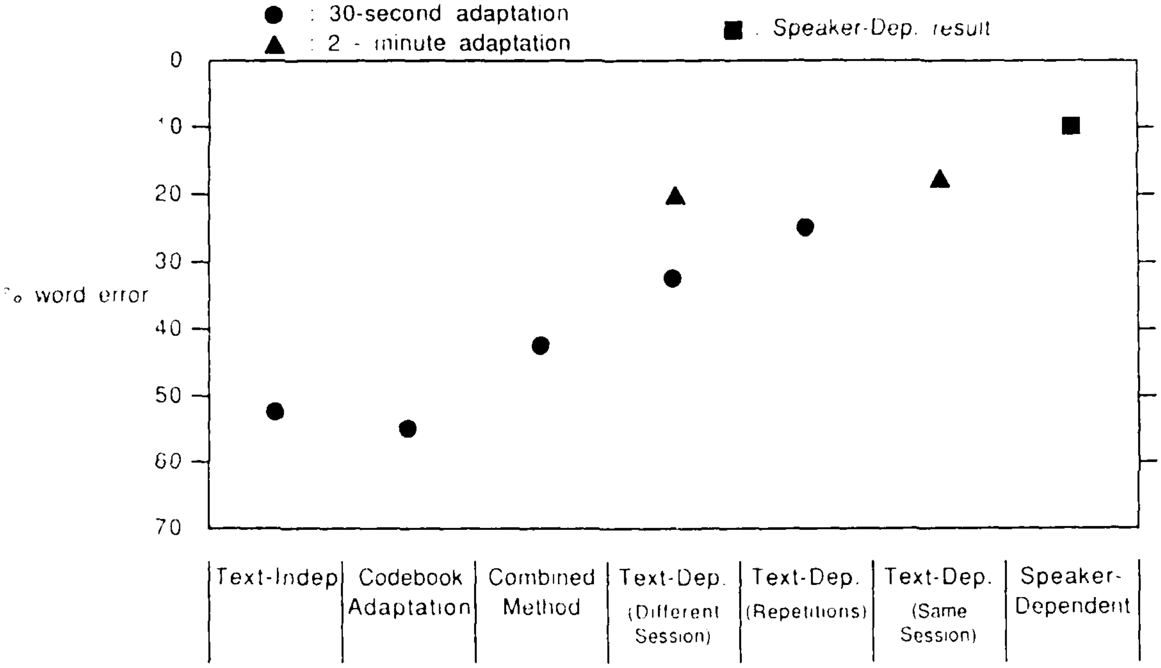

Figure 7: Comparison of Different Methods for Estimating the Transformation Matrix

compare the performance of our algorithms with another existing scheme. This method has been applied in a template-based isolated word recognition system, but here we apply it to an HMM-based continuous speech recognition system. In this algorithm we first align the same sentence spoken by the two different speakers using a DTW algorithm with Euclidean distance between MFCC vectors as the distance measure. Then we replace the decoded value of each codeword of the prototype speaker's codebook with the mean of the input spectra that align to that codeword. Thus, new speech from the input speaker, when quantized using this modified codebook, will tend to get the same codewords as the prototype speaker's speech using the prototype speaker's codebook. Experiments on the 4 speakers show that recognition performance is similar to our previous algorithm (54% word error).

Combined method

Although the codebook adaptation algorithm alone does not perform adequately, it does greatly reduce the quantization error for the new speaker. Therefore, we attempted to use the adapted codebook to quantize the input speech, followed by the original probabilistic text-independent spectral mapping algorithm. Experimental results obtained using this combined procedure show that the word error rate has been significantly reduced to 42%. However, it is still more than 4 times the speaker-dependent error rate.

Text-Dependent Mapping (new algorithm)

In the fourth column we show the performance of the new algorithm using 30 seconds of adaptation speech indicated by the circle. The word error rate (32%) was significantly reduced compared to that of the previous algorithms. Next, we performed some experiments using 2 minutes of adaptation speech. As indicated by the triangle in the fourth column, the word error rate has been significantly reduced to 20% using more adaptation speech. This rate is about two times the speaker-dependent error rate.

Repetition of prototype adaptation speech

To confirm the effectiveness of using multiple repetitions of prototype adaptation speech, we performed a set of experiments using 10 sentences of input speech (30 seconds from the input speaker) and 100 sentences of prototype speech, which is 10 repetitions of the adaptation material. The error rate was 26%, as compared with 32% with only 1 repetition of the 10 sentences.

Session Effect

In all the experiments described above, the adaptation speech was recorded in a different session than the test speech. Here, we performed experiments using 2 minutes of adaptation speech from the same session as the test speech. The error rate was reduced only from 20% to 18% error, which shows that either the different sessions were very similar, or this new algorithm is not very sensitive to the session effect.

Performance on more speakers

To further evaluate the new method we performed experiments on another 7 speakers (4 from TI and 3 from BBN). We used 2 minutes of input speech from the same session as the test speech. Table 7 contains the comparison of speaker-adapted and speaker-dependent performance for each of the 11 speakers. The results show that:

1. The performance difference between 2-minute speaker adaptation and speaker-dependent training are within a factor of two (11.3% versus 7.1%).
2. This algorithm works well for speakers with different dialects than the prototype speaker.
3. The female speakers' models are adapted very well even though the prototype speaker is a male.
4. The difference between the performance of speaker-adapted models and speaker-dependent models is smaller for BBN speakers than for TI speakers. This could be due to several differences, including the fact that the prototype speaker was recorded at BBN.

Future Work

We feel that there are several ways in which the adaptation procedure can be improved.

SPEAKER	RECORDED AT	GENDER	DIALECT	2-MINUTE SPEAKER ADAPTATION	SPEAKER-DEPENDENT TRAINING
CMR	TI	F	Southern	13.5	7.1
JWS	TI	M	Mid South	20.8	5.6
BEF	TI	M	Mid North	15.7	6.6
RKM	TI	M	South	20.3	16.4
TAB	TI	M	Western	7.2	3.2
PGH	TI	M	New England	11.0	6.0
DTD	Ti	F	South	8.6	6.7
DTB	Ti	M	Mid North	9.3	5.4
OK	BBN	M	Mid Atlantic	6.1	6.6
JM	BBN	M	Non-Native	7.6	11.4
FK	BBN	M	Mid North	4.7	3.5
AVERAGE ERROR RATE				11.3	7.1

Table 7: Comparison of Recognition Accuracy of Speaker Adaptation using 2 minutes vs Speaker-Dependent Training using 30 minutes

One simple idea is to have available several prototype speakers, in order to choose the one most appropriate for the new speaker. In one experiment, we used speaker TAB recorded at TI (instead of BBN speaker RS) to be the prototype for input speaker PGH. The performance was improved from the word error rate of 11% to 8%. Thus, the possibility of improving performance by prototype selection remains.

4.3.4 Conclusions

The results above show that the new text-dependent probabilistic spectral mapping algorithm results in significantly better performance than the previous algorithms, and also provides recognition performance which is only two times the word error rate for speaker-dependent training, using 2 minutes of adaptation speech. We believe that further improvement can be achieved by appropriate prototype selection and using multiple repetitions of prototype adaptation speech.

References

1. S.E. Levinson, L.R. Rabiner, and M.M. Sondhi, "Speaker Independent Isolated Digit Recognition Using Hidden Markov Models", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Boston, MA, April 1983, pp. 1049-1052.

2. J.K. Baker, "Stochastic Modeling for Automatic Speech Understanding", in *Speech Recognition*, Raj Reddy, ed., Academic Press, New York, 1975, pp. 521-542, ch. Part Five:systems Organization and Analysis Systems.

3. L.R. Bahl and F. Jelinek, "Decoding for Channels with Insertions, Deletions, and Substitutions with Applications to Speech Recognition", *IEEE Trans. Inform. Theory*, Vol. IT-21, No. 4, July 1975, pp. 404-411.

4. L.R. Bahl, F. Jelinek, and R.L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 2, March 1983, pp. 179-190.

5. L.E. Baum and J.A. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model of Ecology", *Amer. Math Soc. Bulletin*, Vol. 73, 1967, pp. 360-362.

6. J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding", *Proc. IEEE*, Vol. 73, No. 11, November 1985, pp. 1551-1588, Special Issue on Man-Machine Speech Communication.

7. R. Schwartz, Y-L. Chow, M.O. Dunham, O. Kimball, M. Krasner, F. Kubala, J. Makhoul, P. Price, and S. Roucos, "Robust Coarticulatory Modeling for Continuous Speech Recognition", Midterm Report No. 6383, Bolt Beranek and Newman Inc., October 1986, Contract No. N00014-85-C-0279

8. R.M. Schwartz, Y.L. Chow, O.A. Kimball, S. Roucos, M. Krasner, and J. Makhoul, "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, March 1985, pp. 1205-1208, Paper No. 31.3.

9. Y.L. Chow, R.M. Schwartz, S. Roucos, O.A. Kimball, P.J. Price, G.F. Kubala, M.O. Dunham, M.A. Krasner, and J. Makhoul, "The Role of Word-Dependent Coarticulatory Effects in a Phoneme-Based Speech Recognition System", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tokyo, Japan, April 1986, pp. 1593-1596, Paper No. 30.9

10. S. Roucos, R. Schwartz, and J. Makhoul, "Segment Quantization for Very-Low-Rate Speech Coding", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Paris, France, May 1982, pp. 1565-1569.

11. B. -H. Juang and L.R. Rabiner, "Mixture Autoregressive Hidden Markov Models for Speech Signals", *IEEE Trans. Acoust., Speech and Signal Proc.*, Vol. ASSP-33, No. 6, December 1985, pp. 1404-1413.

12. S. Roucos, R. Schwartz, and J. Makhoul, "A Segment Vocoder at 150 B/S", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Boston, MA, April 1983, pp. 61-64.

13. M.M. Sondhi and S.E. Levinson, "Computing Relative Redundancy to Measure Grammatical Constraint in Speech Recognition Tasks", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tulsa, OK, April 1978, pp. 409-412.

14. K. Shikano, K.F. Lee, R. Reddy, "Speaker Adaptation through Vector Quantization", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tokyo, Japan, April 1986, pp. 2643-2646, Paper No. 49.5

15. P.Price, W. Fisher, J. Bernstein, and D. Pallett, "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, New York, NY, April 1988.

16. F. Kubala, Y. Chow, A. Derr, M.Feng, O.Kimball, J. Makhoul, P. Price, J. Rohlicek, S. Roucos, R. Schwartz, and J. Vandegrift, "Continuous Speech Recognition Results of the BYBLOS System on the DARPA 1000-Word Resource Management Database", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, New York, NY, April 1988.