Automatic Generation of Subword Units for Speech Recognition Systems

Rita Singh, Bhiksha Raj, and Richard M. Stern, *Member, IEEE*

***Abstract*—Large vocabulary continuous speech recognition (LVCSR) systems traditionally represent words in terms of smaller subword units. Both during training and during recognition, they require a mapping table, called the dictionary, which maps words into sequences of these subword units. The performance of the LVCSR system depends critically on the definition of the subword units and the accuracy of the dictionary. In current LVCSR systems, both these components are manually designed. While manually designed subword units generalize well, they may not be the optimal units of classification for the specific task or environment for which an LVCSR system is trained. Moreover, when human expertise is not available, it may not be possible to design good subword units manually. There is clearly a need for data-driven design of these LVCSR components. In this paper, we present a complete probabilistic formulation for the automatic design of subword units and dictionary, given only the acoustic data and their transcriptions. The proposed framework permits easy incorporation of external sources of information, such as the spellings of words in terms of a nonideographic script.**

***Index Terms*—Learning, lexical representation, maximum-likelihood, speech recognition, subword units.**

I. Introduction

LARGE vocabulary continuous speech recognition (LVCSR) systems do not usually use whole words as the basic units for classification. There are two reasons for this. First, the vocabulary of these systems typically consists of tens of thousands of words. Even fairly large training corpora typically fail to provide training examples for every word in the vocabulary. Secondly, even large training corpora do not necessarily have enough acoustic examples of all the words in the vocabulary. As such, words which are not seen during training cannot be learned and so can never be recognized. To avoid these problems, LVCSR systems use sound units which are smaller than words as the basic units for classification. Words are translated into sequences of these *subword* units for recognition. Subword units occur much more frequently than words and can therefore be better learned. They also offer the facility of extending the recognition vocabulary to words not seen during training, since new words can always be composed as sequences of these units.

In an LVCSR system, the mapping table which translates words into sequences of subword units is called a *dictionary*. The performance of the LVCSR system depends critically on the choice of the subword units and the accuracy of the dictionary. For example, in a speech transcription task in English, if the sounds represented by “T” and “D” were chosen to be represented by the same subword unit, words differing only in these sounds (like “BAD” and “BAT”) could never be acoustically distinguished. In current large vocabulary systems the dictionary and the subword units are manually designed by experts. This method suffers from the obvious drawback that it cannot be used in the absence of a human expert. Another important consideration is that different modeling paradigms allow different characteristics of sounds to be modeled optimally: static models such as Gaussian mixtures are good for modeling units which are composed of steady-state sounds, whereas sounds with time-varying characteristics such as diphthongs are better modeled by time-varying representations such as hidden Markov models (HMMs). It is clear that a single set of manually defined units may not be coincident with the set that can be best captured by a given model. To some extent the composition of this set may also be influenced by the nature of the acoustic data being recognized. For example, in telephone speech, where much of the high-frequency information is lost, it may not be optimal to use the same variety of fricatives as used for full-bandwidth speech. It may therefore be instructive, if not useful, to devise data-driven automatic methods of deriving the subword units for an LVCSR system.

In this paper, we address the problem of automatically designing the subword units and the dictionary given only a set of acoustic signals and their transcripts. The problem of automatic identification of subword units has been addressed by several researchers in the past [1]–[6]. The earliest efforts treated the problem as one of optimal segmentation and clustering of acoustic examples of words [1], [6]. Other researchers addressed the problem of automatically defining the optimal pronunciation for words in terms of manually defined subword units [7]–[9]. Holter *et al.* [4] and Bacchiani *et al.* [5] have further investigated the problem of automatically determining the basic subword units and the pronunciations of words in terms of these units jointly, using a maximum-likelihood (ML) criterion. All of these methods treat the problem of identifying subword units as one of segmentation and clustering, *albeit* with likelihood as an objective function. Additionally, all of these methods depend on the availability of labeled data, i.e., data where the boundaries of words are marked. Since such databases are usually

Manuscript received September 24, 2001; revised December 11, 2001. This work was supported by the Department of the Navy, Naval Research Laboratory, under Grant N00014-93-1-2005. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dirk van Compernolle.

R. Singh and R. M. Stern are with the Department of Electrical and Computer Engineering and School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

B. Raj is with the Mitsubishi Electric Research Laboratory, Cambridge, MA 02139 USA (e-mail: bhiksha@merl.com).

Publisher Item Identifier S 1063-6676(02)01520-1.

not available, they rely on word boundary labels obtained from speech recognition systems trained with conventional manually designed dictionaries and subword units. Also, while both [5] and [10] do permit the incorporation of simple linguistic knowledge in the estimation of subword units and pronunciations, the use of additional sources of information is not permitted by their framework.

In this paper, we present a complete probabilistic framework for the estimation of subword units and pronunciations, which makes no assumptions about the availability of any *a priori* knowledge or information besides the acoustic training data and their transcripts. While the proposed framework permits the incorporation of diverse external sources of information into the solution in a very simple manner, the existence of these sources of information is not critical to the solution. Thus, while word boundary knowledge can be incorporated if available, it is not explicitly required. We demonstrate how external knowledge can be incorporated into the framework through the usage of statistical correlations between spellings of words and their pronunciations. We use these to improve the consistency of the estimated pronunciations.

In the following section we describe our formulation of the problem. In subsequent sections we present a mathematical formulation for the problem and its solution within a probabilistic framework, followed by some experimental results and our conclusions.

II. Description of the Problem

In this section, we provide the groundwork required for the formulation of the problem of automatically identifying the optimal set of subword units for a given set of acoustic data. For the sake of brevity, in the rest of this paper we will refer to the set of subword units as the phoneset. For the same reason, we will also refer to the subword units themselves as “phones.” This must not be construed to mean that the subword units are *phonetically* motivated in the traditional sense.

The problem itself can be approached from any of three major perspectives:

1) from a modeling perspective, we can try to identify sound classes (which are also the phones) that best fit the training data;
2) from a pattern classification perspective, we can try to identify sound classes that are maximally separable;
3) from a task completion perspective, we can try to find the sound classes that maximize the system’s ability to extract information which is relevant to the completion of a particular task.

In this paper, we choose to approach the problem from the first perspective. The closeness of fit to training data can be quantified by likelihood which, for a data point, is defined to be the value of the probability density function at that point. The higher the likelihood, the better the fit. The assumption that we implicitly make here is that classes which best fit the training data will result in the best classification performance by the LVCSR system on the given acoustic data, as measured by likelihood.

A. Design Based on the ML Criterion

In a dictionary, a phone is merely a symbol. What makes it relevant to the LVCSR system is its consistent usage to represent a particular sound which has a particular distribution or acoustic model associated with it. Therefore, if we find the dictionary in terms of any set of symbols and the acoustic models for those symbols, such that the dictionary and the acoustic models together best fit the data, the ML solution for the problem would have been found.

The problem, therefore, needs to be mathematically formulated as a joint optimization of the dictionary and the acoustic models for the phones, with likelihood maximization as the objective function. This is a very complex problem. While the general aim is to identify the sound classes with the minimum within-class variance, the number of classes to be identified is not known *a priori*. A simple clustering of individual vectors is not sufficient to generate the classes since a sound unit is represented by a sequence of feature vectors, all of which must be considered as one unit. It is not known where, in a given utterance, each of these sequences begin and end. This is complicated by the fact that all sequences of vectors belonging to the same phone need not be of the same length. The typical length of such sequences for a given unit is not known, nor even the distribution of their lengths. Also, the notion of distance between the sound classes is now more complex. In this case a vector sequence belongs to a class, or is closest to a class, only if the statistical model representing that class is more likely to generate that sequence of vectors than the models representing other classes. The list of unknowns is lengthened further by considerations at the word level. In addition to not knowing where each word begins or ends, we also do not know how many phones or classes there are in each word.

The problem therefore must be formulated in such a way as to enable us to identify the vector sequences corresponding to the classes which have to be identified as such, jointly with the generation of a dictionary. As explained in Section I, since the optimality of the phones relates specifically to the statistical models used by the recognizer, this has to be done using the same statistical models and feature set used by the LVCSR system.

III. Formulation and Solution of the Problem

In this section, we present our formulation of the problem and its solution.

Let D_n be a dictionary in terms of a phoneset ϕ_n, where n is the size of the phoneset. The dictionary is a mapping between a set of words and their pronunciations in terms of the phoneset ϕ_n. It can be represented by the set of pronunciations $\{\wp_w\}$, where $\wp_w$ denotes the pronunciation of the word w. Let $\mathbf{A}$ represent the acoustic training data and $\mathbf{T}$ represent their transcriptions. Let λ_n represent the set of n acoustic models for the phoneset ϕ_n. Let E represent any external constraint or source of information about the dictionary and the phoneset that we may consider during solution of the problem. If the transcripts $\mathbf{T}$ are in terms of any nonideographic script, then we may assume that there exist correlations between the spelling and pronunciation(s) of a word. We denote this external knowledge as E_{spel}.

We also assume that in a natural language, certain sequences of sounds are more likely than others while yet others are impossible. We denote this external knowledge as E_{seq}.

As explained in the previous section, the ML formulation of the problem needs to be a joint optimization of the dictionary D_n and the acoustic models λ_n. Assuming that n is known, we formulate the problem of learning the set of subword units and dictionary as the following likelihood maximization:

$$\lambda_n, \mathbf{D}_n = \underset{\Lambda_n, \{\wp_w\}}{\arg\max} P(\mathbf{A}, \{\wp_w\} | \mathbf{T}, n, \Lambda_n, E_{spel}, E_{seq}) \quad (1)$$

where Λ_n is any arbitrary set of n acoustic models and $\{\wp_w\}$ is an arbitrary set of pronunciations representing the dictionary.

Note that this formulation is different from a *true* ML formulation which would be

$$\lambda_n, \mathbf{D}_n = \underset{\Lambda_n, \{\wp_w\}}{\arg\max} P(\mathbf{A} | \{\wp_w\}, \mathbf{T}, n, \Lambda_n, E_{spel}, E_{seq}). \quad (2)$$

However, for a given set of pronunciations $\{\wp_w\}$ the likelihood of the acoustic data is completely determined by the acoustic models Λ_n. Equation (2) could therefore be reduced to

$$\lambda_n, \mathbf{D}_n = \underset{\Lambda_n, \{\wp_w\}}{\arg\max} P(\mathbf{A} | \{\wp_w\}, \mathbf{T}, n, \Lambda_n). \quad (3)$$

This true ML formulation does not utilize external knowledge sources that may constrain the dictionary, if such sources are present. In order to utilize these constraints, it becomes necessary to reformulate the optimization criterion as a *maximum a posteriori* (MAP) estimation of the dictionary.

Equation (1) gives us the optimal dictionary and phoneset for a *given* phoneset size n. However, the optimal value of the variable n itself has to be estimated in this framework. This cannot be estimated on the basis of the likelihood of the training data, since n relates to the total number of parameters in the acoustic models and the likelihood would increase monotonically with increasing n. We can therefore use a set of held-out data $\mathbf{A}_H$, which is not a part of $\mathbf{A}$, to estimate n_{opt}, the optimal value of n

$$n_{opt} = \underset{n}{\arg\max} L(\mathbf{A}_H | n) \equiv \underset{n}{\arg\max} P(\mathbf{A}_H | \mathbf{D}_n, \lambda_n) \quad (4)$$

where $\mathbf{D}_n$ and λ_n are the optimal dictionary and acoustic models for the given n, as obtained from (1). Alternatively, n can be chosen to optimize the recognition accuracy obtained with D_n and λ_n on the heldout data.

IV. Solution of the ML Formulation for the Joint Estimation of Dictionary and Phone Set

The function $P()$ in (1) is not easy to solve directly for a global optimum. It must be decomposed into simpler components to facilitate solution.

A. Divide-and-Conquer Strategy

It can be shown (see Appendix A) that (1) can be decomposed into two equations which, when solved iteratively, are guaranteed to converge to a locally optimal solution. These equations are

$$\begin{aligned} \lambda_n^i = \underset{\Lambda_n}{\arg\max} \{ & P(\mathbf{A} | \mathbf{T}, \mathbf{D}_n^i, n, \Lambda_n, E_{spel}, E_{seq}) \\ & \cdot P(\mathbf{D}_n^i | \mathbf{T}, n, \Lambda_n, E_{spel}, E_{seq}) \} \end{aligned} \quad (5)$$

$$\mathbf{D}_n^{i+1} = \underset{\{\wp_w\}}{\arg\max} P(\{\wp_w\} | \mathbf{A}, \mathbf{T}, n, \lambda_n^i, E_{spel}, E_{seq}) \quad (6)$$

where the superscript i represents the iteration number. To solve these equations we first fix the phoneset size n and initialize the dictionary in some simple manner (dictionary initialization is discussed in a Section IV-E). Then, assuming that the dictionary is given, we find the best acoustic models. In the next step we use these acoustic models and find the best dictionary, and so on.

Equations (5) and (6) can be further simplified by noting that the knowledge of n, the size of the phoneset, is implicit in the knowledge of the dictionary $\mathbf{D}_n^i$. Similarly, once λ_n^i is known, n is implicitly known. The variable n therefore need not appear explicitly in the equations. A second simplifying consideration is that in the absence of acoustic data relating the acoustic models to the dictionary, the two are independent. Hence the term $P(\mathbf{D}_n^i | \mathbf{T}, n, \Lambda_n, E_{spel}, E_{seq})$ does not affect the solution of (5). Further simplification of (5) can be done by noting that the probability of the acoustic data $\mathbf{A}$ depends only on the dictionary and the statistical models for the data and can be assumed to be independent of any phonemic or spelling constraints. In the light of these considerations (5) and (6) reduce to

$$\lambda_n^i = \underset{\Lambda_n}{\arg\max} P(\mathbf{A} | \mathbf{T}, \mathbf{D}_n^i, \Lambda_n) \quad (7)$$

$$\mathbf{D}_n^{i+1} = \underset{\{\wp_w\}}{\arg\max} P(\{\wp_w\} | \mathbf{A}, \mathbf{T}, \lambda_n^i, E_{spel}, E_{seq}). \quad (8)$$

We refer to (7) and (8) as the *model update* and the *dictionary update* equations, respectively. In the following paragraphs we explain how these can be solved by reapplying the divide-and-conquer strategy.

B. Solution of the Model Update Equation

The model update equation (7) is the ML solution for the statistical models of the phones for a corpus of training data, given a dictionary. The method used to solve this equation would be dependent on the actual statistical model used. Typically the solution would involve the use of an expectation maximization (EM) algorithm [11]. When the statistical models are HMMs, the Baum–Welch algorithm [12] may be used to solve for λ_n^i.

The dictionary update equation (8), on the other hand, represents a maximum *a posteriori* estimate for the dictionary, $\mathbf{D}_n^{i+1}$ given the statistical models for the phones, λ_n^i, the training corpus and the external constraints. This equation is again too complex to solve directly and must be simplified for the purpose.

C. Simplification of the Dictionary Update Equation

In order to simplify (8), we introduce a *word-segmentation* variable seg_w, which represents any possible segment of the speech signal that may correspond to the given word w. Before we show how this variable can be used to simplify the equation for the dictionary update, there are some points that must be considered in relation to the nature of the variable. Since at this point the word boundaries in any particular utterance are not known, the only condition that we can impose on them is that the number of word segments in an utterance must be equal to the number of words in it. Since an utterance consists of a finite and discrete number of frames or samples, there are clearly a finite but large number of ways of segmenting an utterance into a specified number of words. The variable seg_w alludes to each of these possible segmentations corresponding to any word w. The set $\{seg_w\}$ refers to all possible word segmentations for the word w, not all of which may be close to the *true* segmentation.[1]

The word-segmentation variable can be introduced into (8) as a null-factor that leaves it unchanged

$$\mathbf{D}_n^{i+1} = \arg\max_{\{\wp_w\}} \sum_{\{seg_w\}} P(\{\wp_w\}, \{seg_w\}| \mathbf{A}, \mathbf{T}, \lambda_n^i, E_{spel}, E_{seq}) \quad (9)$$

where the right-hand side of the equation is summed over *every* possible segmentation of the training data into the sequence of words given by $\mathbf{T}$. We now make a convenient approximation: since the actual number of possible word segmentations for any corpus of training data is very large, we assume that only the *best* word segmentation affects the contents of the optimal dictionary and estimate it jointly with the dictionary. We thus jointly optimize $\{\wp_w\}$ and $\{seg_w\}$ and approximate the optimal dictionary with the corresponding optimal value of $\{\wp_w\}$

$$\mathbf{D}_n^{i+1} = \arg\max_{\{\wp_w\},\{seg_w\}} P(\{\wp_w\}, \{seg_w\}| \mathbf{A}, \mathbf{T}, \lambda_n^i, E_{spel}, E_{seq}). \quad (10)$$

While this may appear to be more difficult to solve in general situations than (8), it actually simplifies the problem. We can use the constructs proved in Appendix A to reapply the divide-and-conquer strategy. Following this, (10) can be decomposed into two equations again, which when iteratively solved are guaranteed to converge to a locally optimal solution

$$\{seg_w\}^j = \arg\max_{\{seg_w\}} P\big(\{seg_w\}|\mathbf{D}_n^{i+1,j}, \mathbf{A}, \mathbf{T}, \lambda_n^i, E_{spel}, E_{seq}\big) \quad (11)$$

$$\mathbf{D}_n^{i+1,j+1} = \arg\max_{\{\wp_w\}} P\big(\{\wp_w\}|\{seg_w\}^j, \mathbf{A}, \mathbf{T}, \lambda_n^i, E_{spel}, E_{seq}\big). \quad (12)$$

We refer to (11) and (12) as the *word-segmentation update* and the *word-segmentation based dictionary update* equations, respectively. The variables i and j represent iteration numbers.

The procedure suggested by the above equations is to fix the dictionary first and find the most likely word segmentations. The word segmentations are subsequently used to find the best dictionary. In the following subsections we will further show how to simplify and solve (11) and (12).

1) Simplification of the Word-Segmentation Update Equation: Equation (11) can be rewritten as

$$\{seg_w\}^j = \arg\max_{\{seg_w\}} \left\{ P\big(\mathbf{A}|\{seg_w\}, \mathbf{T}, \mathbf{D}_n^{i+1,j}, \lambda_n^i, E_{spel}, E_{seq}\big) \cdot \frac{P(\{seg_w\}|\mathbf{T}, \mathbf{D}_n^{i+1,j}, \lambda_n^i, E_{spel}, E_{seq})}{P(\mathbf{A}|\mathbf{T}, \mathbf{D}_n^{i+1,j}, \lambda_n^i, E_{spel}, E_{seq})} \right\}. \quad (13)$$

If we assume that all valid word segmentations of the training corpus are equally likely when not conditioned on acoustic evidence,[2] (13) gets simplified to

$$\{seg_w\}^j = \arg\max_{\{seg_w\}} P(\mathbf{A}|\{seg_w\}, \mathbf{T}, \mathbf{D}_n^{i+1,j}, \lambda_n^i, E_{spel}, E_{seq}). \quad (14)$$

This equation can be solved using the Viterbi algorithm [12]. Note that if *a priori* probabilities were available for seg_w, they could be incorporated into (13) and the assumption of equiprobable segmentations would not be required.

2) Simplification of the Word-Segmentation Based Dictionary Update Equation: In (12), the set $\{\wp_w\}$ represents the jointly optimal set of pronunciations of all the words in the dictionary. Joint optimization of all the pronunciations in the dictionary is a reasonable requirement in light of the fact that pronunciations of words in a dictionary are not independent of each other. They are correlated, and we expect words with similar spellings to have similar pronunciations. However, jointly optimizing for the pronunciations of what could be thousands of words in a dictionary is a very complex problem. Equation (12) needs to be simplified further.

The simplifying assumption that we make is based on the observation that within any given iteration of (11) and (12), the actual boundaries of all the words in the training corpus are known, once $\{seg_w\}^j$ is known. While these are possibly not the best or the true word boundaries, the fact that they are *known* allows us to now make the approximation that the pronunciations for all the words in the dictionary need not be *jointly* optimized. Instead, it is sufficient to optimize the pronunciation of each word in the dictionary separately. Therefore, instead of

$$\mathbf{D}_n^{i+1,j+1} = \{\wp_w\}^{\max} \quad (15)$$

we consider it sufficient to obtain

$$\mathbf{D}_n^{i+1,j+1} = \{\wp_w^{\max}\} \quad (16)$$

where

$$\wp_w^{\max} = \arg\max_{\wp_w} P(\wp_w|A_w, \lambda_n^i, E_{spel}, E_{seq}) \quad (17)$$

[1] By *true* segmentation, we refer to a segmentation that would be obtained by an ideal recognition system that has been trained on infinite data and represents the true distribution of the speech.

[2] In reality, the probability of any word segmentation would be dependent on the parameters of the underlying Markov chain. However, we do not expect the assumption of equiprobable segmentations to affect the solution greatly. It merely facilitates the usage of the Viterbi algorithm to estimate $\{seg_w\}^j$. The estimation would otherwise be tedious.

where A_w refers to the acoustic data corresponding to all instances of the word w. Equation (17) however requires us to search over every possible pronunciation $\wp_w$ to identify $\wp_w^{\max}$. For any word w, there is a large number of possible pronunciations in the absence of any constraint. In the limiting case where the acoustic model is able to associate only one feature vector with each phone, for n phones and m feature vectors present in a considered segmentation for a word, the number of possible pronunciations can be as large as n^m. Direct evaluation of (17) is clearly infeasible. Solutions do exist for the ML version of (17) [2]. However, even those solutions are computationally expensive and have to be reduced by considering only a subset of possible pronunciations as in [4].

To make the problem more tractable, we confine the pronunciations considered in (17) to only the set of pronunciations evidenced in the acoustic training corpus, after expanding that set a little using a graph as explained below:

For any *single* instance w_k of a word w with corresponding acoustic data A_{w_k}, it is easy to obtain

$$\wp_{w_k}^{\max} = \arg\max_{\wp_w} P(A_{w_k} | \wp_w, \lambda_n^i) \tag{18}$$

using the Viterbi algorithm. We obtain $\wp_{w_k}^{\max}$ for every instance of the word w_k in the training set, resulting in a set of pronunciations $\{\wp^{\max}\}_w$ for the word w. This set of pronunciations can be collapsed into a graph [13], [14] as shown in Fig. 1.

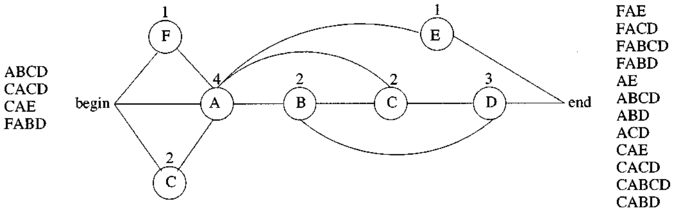


Fig. 1. Graph constructed with four hypothetical pronunciations of a word, listed on the left. Once constructed, the graph permits new paths from begin to end, and thus can generate twelve pronunciations for the word. These are listed on the right.

As can be seen from this figure, the graph enables us to generate many more putative pronunciations for the word than the original set of pronunciations $\{\wp^{\max}\}_w$ that were used to create the graph. In Fig. 1, four hypothetical pronunciations for a word are collapsed into a single graph. These are listed on the left of the graph. The weight associated with any node is proportional to the number of times the node has been visited in this set of four pronunciations. This is indicated on the top of each node in the graph. On the right of the graph in Fig. 1 are listed 12 pronunciations which can now be generated from the graph. Following the same procedure, we expand the set of pronunciations $\{\wp^{\max}\}_w$ for each unique word in the corpus to a set of pronunciations $\{\wp_w\}_{w_{graph}}$ by tracing every possible path through this graph [15]. We then finally restrict our search for the optimal pronunciation in (17) to this set of pronunciations.

If we include the corresponding pronunciation from $\mathbf{D}_n^{i+1,j}$ in $\{\wp_w\}_{w_{graph}}$, the most likely pronunciation in $\{\wp_w\}_{w_{graph}}$ is guaranteed to be *at least* as likely as the pronunciation in $\mathbf{D}_n^{i+1,j}$, thereby guaranteeing a nondecreasing likelihood for every iteration. Equation (17) now becomes

$$\wp_w^{\max} = \arg\max_{\wp_w \in \{\wp_w\}_{w_{graph}}} P(\wp_w | A_w, \lambda_n^i, E_{spel}, E_{seq}). \tag{19}$$

This equation can now be simplified as follows:

$$\begin{aligned} &\arg\max_{\wp_w \in \{\wp_w\}_{w_{graph}}} P(\wp_w | A_w, \lambda_n^i, E_{spel}, E_{seq}) \\ &= \arg\max_{\wp_w \in \{\wp_w\}_{w_{graph}}} \\ &\quad \cdot \frac{P(A_w | \wp_w, \lambda_n^i, E_{spel}, E_{seq}) P(\wp_w | \lambda_n^i, E_{spel}, E_{seq})}{P(A_w | \lambda_n^i, E_{spel}, E_{seq})}. \end{aligned} \tag{20}$$

To simplify this, we note that $P(A_w | \lambda_n^i, E_{spel}, E_{seq})$ is not a function of $\wp_w$. It can also be safely assumed that the probability of a phone sequence $\wp_w$ is not dependent on λ_n^i in the absence of the acoustic data A_w. Equation (19) therefore reduces to

$$\begin{aligned} \wp_w^{\max} = \arg\max_{\wp_w \in \{\wp_w\}_{w_{graph}}} &\{P(A_w | \wp_w, \lambda_n^i, E_{spel}, E_{seq}) \\ &\cdot P(\wp_w | E_{spel}, E_{seq})\}. \end{aligned} \tag{21}$$

Once a specific phone sequence is *given*, the external constraints become inconsequential, since they apply specifically to phone sequences. We therefore have

$$P(A_w | \wp_w, \lambda_n^i, E_{spel}, E_{seq}) = P(A_w | \wp_w, \lambda_n^i). \tag{22}$$

Using Bayes' rule, we also have

$$\begin{aligned} &P(\wp_w | E_{spel}, E_{seq}) \\ &\quad = \frac{P(E_{seq} | E_{spel}, \wp_w) P(E_{spel} | \wp_w) P(\wp_w)}{P(E_{spel}, E_{seq})}. \end{aligned} \tag{23}$$

We make the reasonable assumption that the phone-sequence constraints E_{seq} are characteristic of the phonetic nature of the language, and that they are independent of the script used for the language or the manner in which one chooses to spell words. As a result (23) becomes

$$P(\wp_w | E_{spel}, E_{seq}) = \frac{P(E_{seq} | \wp_w) P(E_{spel} | \wp_w) P(\wp_w)}{P(E_{spel}, E_{seq})} \tag{24}$$

which, through Bayes' rule, can be simplified to

$$\begin{aligned} &P(\wp_w | E_{spel}, E_{seq}) \\ &\quad = \frac{P(\wp_w | E_{seq}) P(E_{seq}) P(\wp_w | E_{spel}) P(E_{spel})}{P(\wp_w) P(E_{spel}, E_{seq})} \end{aligned} \tag{25}$$

where $P(\wp_w)$ is the *a priori* probability of the phone sequence $\wp_w$. If at any point we assume that in the absence of any other information all phone sequences in $\{\wp_w\}_{w_{graph}}$ are equally likely, this term becomes a constant. $P(E_{spel})$, $P(E_{seq})$, and $P(E_{spel}, E_{seq})$ are all independent of $\wp_w$ and are therefore inconsequential in (25). Hence, using (22) and (25) in (21), we get

$$\begin{aligned} \wp_w^{\max} = \arg\max_{\wp_w \in \{\wp_w\}_{w_{graph}}} &\{P(A_w | \wp_w, \lambda_n^i) P(\wp_w | E_{seq}) \\ &P(\wp_w | E_{spel})\}. \end{aligned} \tag{26}$$

$P(A_w | \wp_w, \lambda_n^i)$ is the likelihood of the observed acoustic data for the word for the phone sequence $\wp_w$. If the statistical

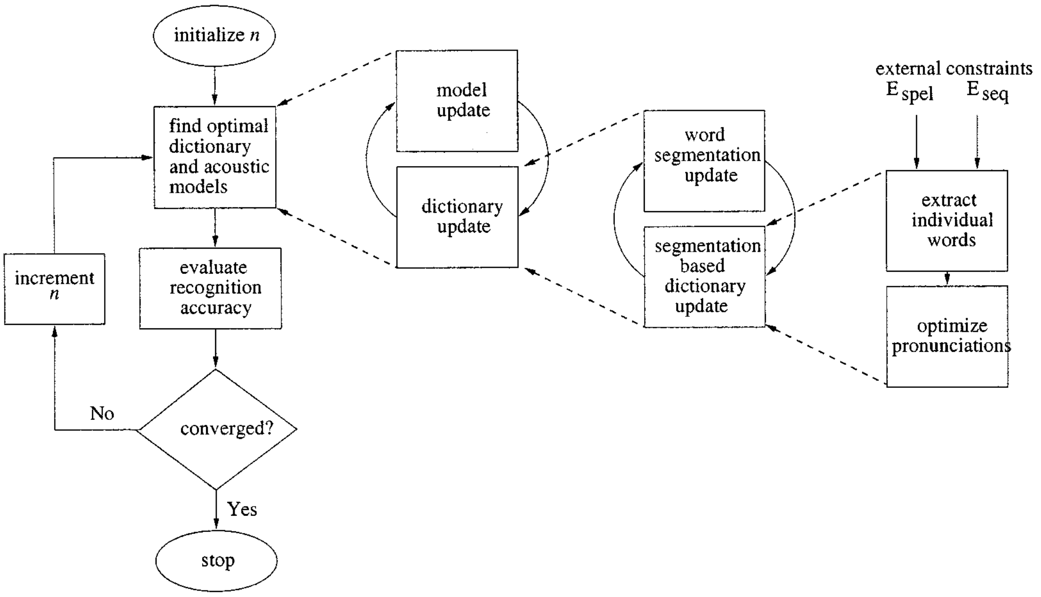


Fig. 2. Algorithm suggested by (1), (8), (9), (11), (12), (14), and (27) for the automatic generation of subword units and dictionary.

models for the phones are HMMs, this likelihood can be easily obtained using either the forward or the backward pass of the Baum–Welch algorithm [12] on all instances of the word (the product of the likelihoods of the individual instances of the word gives us the total likelihood for A_w). $P(\wp_w|E_{seq})$ is the probability of the phone sequence $\wp_w$ given the constraints E_{seq}. If E_{seq} takes the form of rules this would simply result in a 1/0 binary value for $\wp_w$ indicating whether the given rules permit $\wp_w$ or not, as in a word-pair language model. If E_{seq} is a statistical model, e.g., an N-gram model [16], this evaluates the probability of the phone sequence $\wp_w$ on the model. The spelling constraints, E_{spel}, are easily imposed. If these are statistical (a statistical model relating spellings to sequences of phones can be computed, e.g., using techniques described in [17]), $P(\wp_w|E_{spel})$ gives us the probability of the phone sequence $\wp_w$ computed on the spelling to pronunciation model E_{spel}.

Using (16) and (26), the word-segmentation based dictionary update equation for $\mathbf{D}_n^{i+1,\,j+1}$ can now be written as

$$\begin{aligned}\mathbf{D}_n^{i+1,\,j+1} &= \{\wp_w^{\max}\}\\ &= \left\{\begin{aligned}&\mathop{\arg\max}_{\wp_w \in \{\wp_w\}_{w_{graph}}} \{P(A_w|\wp_w,\,\lambda_n^i)\\ &\qquad P(\wp_w|E_{seq})P(\wp_w|E_{spel})\}\end{aligned}\right\}. \qquad (27)\end{aligned}$$

Equations (11) and (12) are to be iterated until $P(\mathbf{D}_n^{i+1,\,j+1},\,\{seg_w\}^j|\mathbf{A},\,\mathbf{T},\,\lambda_n^i,\,E_{spel},\,E_{seq})$ converges. In practice, we test for the convergence of $P(\mathbf{D}_n^{i+1,\,j+1}|\{seg_w\}^j,\,\mathbf{A},\,\mathbf{T},\,\lambda_n^i,\,E_{spel},\,E_{seq})$. The converged value of $\mathbf{D}_n^{i+1,\,j+1}$ gives us $\mathbf{D}_n^{i+1}$ in (8).

The model update (7) and dictionary update (8) steps must be iterated until (1) converges. In practice we iterate the steps until the recognition accuracy on a *heldout* set of data converges.

As a summary, the sequence of steps involved in the solution of (1) are shown in Fig. 2. This figure presents the algorithm suggested by (1), (7), (8), (11), (12), (14), and (27) in the form of a flow chart. In Fig. 2, we begin with an initial phoneset size n and an initial dictionary with any n symbols as the phoneset. We then iterate the dictionary update and model update steps. The dictionary update is in turn iteratively done by using a fixed dictionary to find the best word segmentation, and using the word segmentation to find the best dictionary. Since the pronunciations of all the words in the dictionary cannot be jointly optimized, we accomplish this piecewise by optimizing the pronunciation of each word independently. In the process we use external constraints that we learn in an unsupervised manner [17] to ensure that the pronunciations stay consistent. Once we have the best dictionary and acoustic models we test them on a held out set. If the recognition accuracy is higher than it was with the previous phoneset size, we increase the phoneset size by splitting the phones.

D. Estimating n, the Size of the Phoneset

So far, we have assumed that n, the size of the phoneset, is given. In reality it must be determined empirically. As mentioned in Section III, the phoneset size n cannot be determined on the basis of the likelihood of the training data. We therefore estimate n as

$$n_{opt} = \mathop{\arg\max}_{n}\{Recog(\mathbf{A}_H|\mathbf{D}_n,\,\lambda_n)\} \qquad (28)$$

where $Recog(\mathbf{A}_H|\mathbf{D}_n,\,\lambda_n)$ refers to the recognition accuracy on a set of heldout data $\mathbf{A}_H$, which has not been included in the training. $\mathbf{D}_n$ and λ_n are the optimal dictionary and acoustic models for phoneset size n.

Equation (28) requires the estimation of $\mathbf{D}_n$ and λ_n for every value of n. We begin with a small value for n and increase it gradually until the value of n that maximizes $Recog()$ is found. At every stage the phoneset size is increased in a manner that maximizes the increase in likelihood due to increasing the phoneset size. To accomplish this we cluster the data corresponding to each phone (obtained through phone segmentations derived using the current acoustic models and dictionary) into two clusters and identify the phone for which the clustering resulted in the highest increase in likelihood. The likelihood is measured assuming Gaussian distributions for the data clusters.

Each cluster for the identified phone is now a new (relabeled) phone. Thus with each such split, the phoneset size is increased by one. The relabeled phone sequences replace the original (unsplit) phone labels in the dictionary. The algorithm can then proceed using the new increased phoneset size. Note that if we desire to increase the phoneset size by more than one at any given stage, the splitting can be done for a list of phones which result in high likelihood increases after clustering.

It is important that the clustering technique and the criterion used be consistent with the model used by the recognizer. For example, if the recognizer is HMM-based, the clustering would have to be such that the likelihoods of the clusters on HMMs trained from the segments in the cluster, are maximized. As an example, we may use the hybrid clustering technique described in [18].

E. Initialization of the Dictionary

The algorithm requires the initialization of the dictionary at the outset. Any reasonable heuristically derived initialization is sufficient. For example, if we assume that the script used to transcribe the acoustic training data is nonideographic, one possible way to initialize the dictionary would be to use the alphabet as the initialization: if words are transcribed using the English alphabet (irrespective of language), we could use the alphabet as a phoneset to initialize the pronunciations of all words in the dictionary. Alternatively, we could initialize any word with a sequence of repetitions of a single symbol, the sequence length being approximately proportional to the length of the word. This is the most noncommittal initialization possible, since it is minimally dependent on the consistency of the script of the language. The only assumption made here would be that the length of the spelling of a word is roughly proportional to the number of phones in the word. As the algorithm progresses, the size of the phone set can be increased using cluster-based splitting as described in Section IV-D.

V. Experimental Results

A pilot test for the algorithm for automatic generation of phoneset and dictionary was performed using the resource management (RM) database [19]. The training corpus consisted of 2880 utterances, comprising 2.74 h of acoustic signals. The training set covered a vocabulary of 987 words.

Acoustic models built using the automatically generated phoneset and dictionary were tested on a heldout RM test set, which consisted of 1600 utterances comprising 1.58 h of acoustic signals. The vocabulary of this set was 991 words, four of which were not seen during training. The CMU SPHINX-III speech recognition system was used for acoustic modeling. All acoustic models were semi-continuous five-state HMMs [20] sharing 256 Gaussian densities.

The words in the heldout test set which were not part of the training set were not included in the recognition lexicon in this experiment, since no pronunciations were available for them. However, the generation of pronunciations for new words is not a major problem since several widely used algorithms exist that can learn the relationships between spellings and pronunciations from an existing dictionary and derive pronunciations for new words, e.g., [21]. Most such tools make no explicit assumptions about the nature of the phonetic units and merely treat them as symbols. It is therefore reasonable to expect that they would work as well with automatically learned sound units as with phonetically motivated ones.

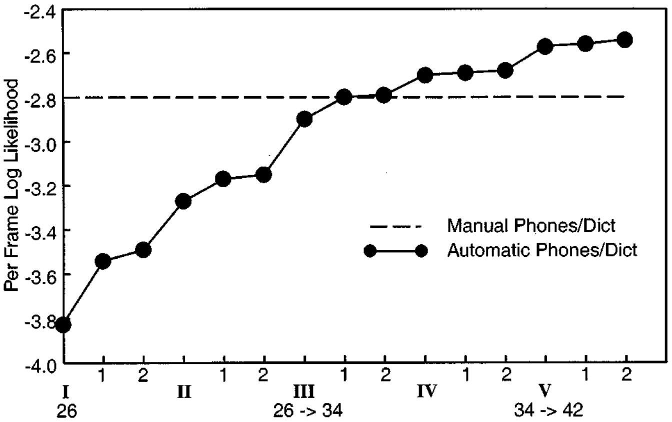


Fig. 3. Likelihood versus iteration for the automatic phone generation experiment with RM. The model update steps are indicated by Roman numerals (I, II, …), and the dictionary update steps are indicated by Arabic numerals (1, 2, …). The phoneset expansions are indicated as $a \longrightarrow b$, where a refers to the size of the phoneset prior to splitting and b refers to the size of the phoneset after splitting.

A baseline was first established using the CMU dictionary (CMUdict) [22], which is a standard, manually crafted dictionary that uses a set of 50 manually designed phonetic units. Although CMUdict has multiple pronunciations for every word, only the most frequently used pronunciations in the RM task were used for the baseline. Also, while the RM task has a very constrained linguistic structure, the experiments took minimal advantage of it. A simple bigram language model was used for the experiment and the weight given to the language model was set to be very small in order to emphasize the contribution of the acoustic models to the recognition. Note that as a result of this, the word error rates reported on the RM task in this paper are higher than the best obtainable by the SPHINX-III system.

For this experiment the dictionary was initialized with the script of the language, where the pronunciation of each word was simply assumed to be the sequence of alphabetical characters which constituted the spelling of the word. The initial dictionary thus had a 26-symbol phoneset corresponding to the English alphabet.

Figs. 3–5 show the results obtained during various stages of the experiment. In these figures the model update steps are indicated by Roman numerals (I, II, …), and the dictionary update steps are indicated by Arabic numerals (1, 2, …). At each model update step, multiple iterations of Baum–Welch were carried out until the likelihood on the training data converged to a local maximum. The phoneset expansions are indicated as $a \longrightarrow b$, where a refers to the size of the phoneset prior to splitting and b refers to the size of the phoneset after splitting. There were two dictionary update steps for each model update step, and the phoneset was split twice, increasing in size from 26 to 34 and subsequently to 42 phones.

We observe in Fig. 3 that the likelihood of the training data increases monotonically with the model and dictionary updates. The likelihood becomes equal to the baseline obtained using the manually designed dictionary and phoneset with only 34 phones, and becomes higher than the baseline with 42 phones. We note in Fig. 3 that the likelihood obtained with the CMUdict, which has 50 phones, is *lower* than that obtained with the 34-phone automatically generated dictionary. This indicates that as far as our criterion of maximum likelihood is concerned, the proposed algorithm is successful in giving us a phoneset which results in distributions that better fit the acoustic training data compared to the phoneset in the CMUdict.

Fig. 4, on the other hand, shows that the best word error rate obtained on the test set is for 34 phones, and that the higher training likelihoods seen in Fig. 3 do not translate to greater recognition accuracy on the test set. However, on the training set the word error rate continues to decrease with increasing phoneset size and training likelihood. This indicates that increasing the phoneset size beyond 34 phones leads to overfitting of the models to the training data, and thus poorer generalizability to the test data, further leading to poorer word error rates on the test set. This also indicates that training set likelihoods are not reliable indicators of the test word error rates.

Fig. 4 raises the valid question that if the word error rate increase with 42 phones is a result of overparametrization, then for the CMUdict which has 50 phones, and therefore even more parameters, the word error rates should be even higher. However, in the case of the CMUdict the larger number of parameters does not result in overfitting as seen from the likelihoods in Fig. 3. This can probably be attributed to the vast amount of human knowledge which has gone into designing the CMUdict. Looking at the trends in Fig. 4 we might, nevertheless, speculate that even for the manually designed phones, 50 may not be the optimal size of the phoneset for the current RM task. The optimal size of the phoneset may depend on the amount of training data.

Fig. 5 shows how the word segmentations for a sample utterance in the training data set evolve as the phoneset and dictionary evolve. The top row of text in the figure shows the actual, manually demarcated, word boundaries. The second row shows the segmentations obtained with the baseline system using manually designed phones and dictionary. The subsequent rows show word segmentations at various stages in our experiment. The stages are labeled on the ordinate according to our specified convention mentioned earlier in this section. We observe from these rows that after just a few iterations the word segmentations converge to specific values which are congruent with the word segmentations obtained using the CMUdict.

The best 34-symbol phoneset and the corresponding dictionary were also evaluated by building context dependent (triphone) semi-continuous HMMs with 2000 tied states. For comparison, context-dependent models with 2000 tied states were also built for the baseline system. The SPHINX-III speech recognition system uses decision trees built using pre-defined phonetic classes called "linguistic questions" for building tied-state context dependent models. While manually designed linguistic questions were available for the baseline system, these were obviously not available for the automatically designed phoneset. For a fair comparison, therefore, the linguistic questions were automatically generated in both cases using the procedure described in [18]. It has been demonstrated in [18] that automatically designed linguistic questions result in word error rates that are comparable to those obtained using manually designed questions. Table I lists the word error rates obtained in this experiment. We note here that although context-dependent HMMs with 2000 tied states have many more parameters than context-independent models with only 42 phones, they result in much lower word error rates. This is because the context specificity in context-dependent models introduces an implicit phone-level grammar which, when appropriately modeled, more than compensates for the loss in generalizability due to overfitting. This structuring is not available for context-independent units.

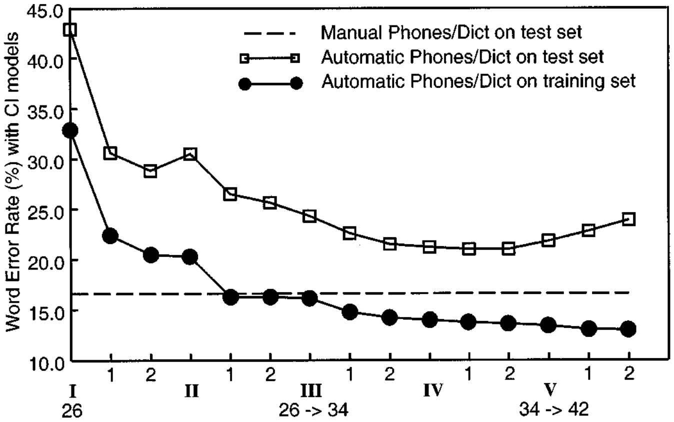


Fig. 4. Word error rate versus iteration for the automatic phone generation experiment with RM. The model update steps are indicated by Roman numerals (I, II, ...), and the dictionary update steps are indicated by Arabic numerals (1, 2, ...). The phoneset expansions are indicated as $a \rightarrow b$, where a refers to the size of the phoneset prior to splitting and b refers to the size of the phoneset after splitting.

We would like to emphasize here that the results described in this section are from a pilot experiment designed to demonstrate the applicability of the algorithm, rather than to generate the optimal phoneset for the RM database. Our implementation of the pilot experiment suffers from several shortcomings due to logistic constraints. Only one pronunciation was generated for each word. Multiple pronunciations can be generated following the procedure outlined in Appendix B, if desired. This would however involve a large amount of computation to estimate the pronunciations for any word. We note also that only the single most likely phone sequence for each instance of a word w was used to generate the graph that was used to produce $\{\wp_w\}_{w_{graph}}$. N-best pronunciations could have been generated instead, and used for the graph. This would increase the size of $\{\wp_w\}_{w_{graph}}$, resulting in a more optimal search for the pronunciation of each word. Context-independent phone models were used throughout the phoneset generation process. Context-dependent models generally result in better recognition accuracies, and the use of context-dependent models may therefore be expected to result in a better dictionary.

We would like to add a few words of caution here: in our experiment the acoustic models are initialized using a flat initial-

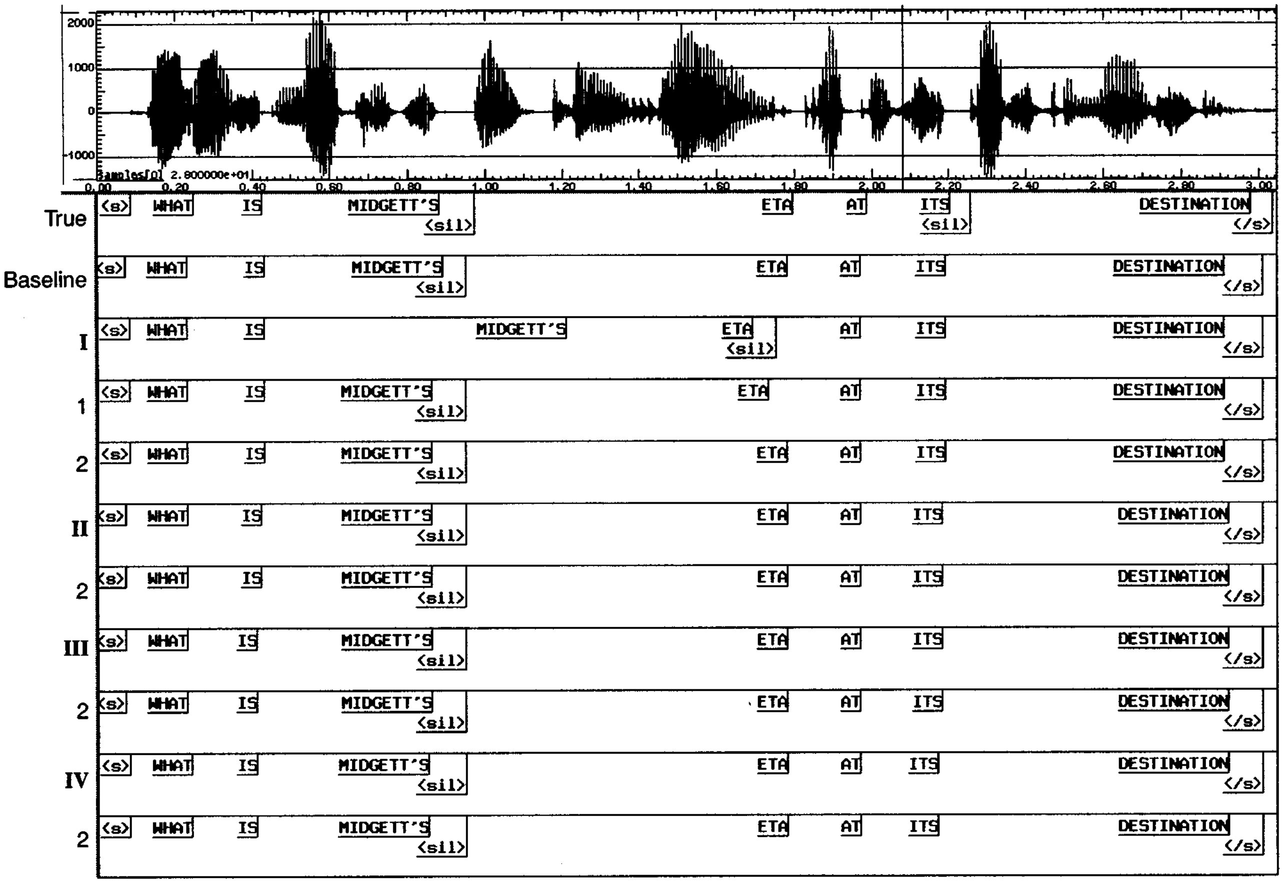


Fig. 5. Evolution of the word segmentations for a sample utterance in the training corpus, as the phoneset and dictionary evolve. The model and dictionary update steps are indicated on the left of the figure and progress vertically from top to bottom.

TABLE I
WORD ERROR RATES OBTAINED WITH MANUAL AND AUTOMATIC SUB-WORD UNITS FOR THE RESOURCE MANAGEMENT DATABASE WITH CONTEXT-DEPENDENT SEMI-CONTINUOUS FIVE-STATE HIDDEN MARKOV MODELS

No.of phones	Design of phoneset/lexicon	wer%
50	manual	9.2
34	automatic	12.6

ization scheme, whereby all state distributions are initially set to be identical to the global distribution of the data. This is likely to be far more effective when training utterances are short. Utterance boundaries implicitly incorporate human knowledge about the boundaries within which a certain set of words occur. As training utterances become longer this knowledge is reduced as fewer boundaries are available for any given amount of training data, adversely affecting the outcome of the algorithm. Secondly, if the script used to represent the language is ideographic, spelling to phone mappings cannot be obtained. As a result words that are poorly represented in the training set may be badly translated into phones. Also the addition of new words into the dictionary may not be possible.

VI. DISCUSSION AND CONCLUSION

In this paper, we have presented an ML formulation for the problem of automatic generation of subword units and dictionary, and explained how a divide-and-conquer strategy can be used to arrive at the solution. Through pilot experiments using the RM database, we have demonstrated the applicability of the solution proposed. The framework we have presented permits us to work in a situation where the only resources available are the acoustic data and their transcriptions. Where additional sources of information are available, it also allows us to incorporate these into the solution easily.

The pilot experiment demonstrated the success of the algorithm in terms of the objective criterion which was maximized. However, the automatically generated subword units and dictionary resulted in models which performed worse than the manually designed subword units and dictionary. Any phoneset and dictionary generated by a human expert virtually uses the knowledge derived from experience with hundreds, even thousands, of hours of speech. It also uses other forms of consciously or subconsciously acquired knowledge. The manually designed phoneset is therefore expected to be highly generalizable. In comparison, the automatically derived phone set and dictionary used only 2.7 h of speech in our experiments. No other source of information was used. The word error rates obtained in our pilot experiments were influenced by this fact.

Although it is obvious that if other sources of information are available they *should* be used to condition the phone generation process, human knowledge of the kind used in the design of phoneset and dictionary for a language is not currently completely quantifiable. It can be argued that until we find ways of doing so, carefully designed manual phonesets and dictionaries will always outperform automatic ones, especially as the complexity (i.e., vocabulary, perplexity, variety of environmental conditions and speaking styles, etc.) of the underlying task increases. The size of the training corpus will also continue to limit the quality of the automatically learned phones. Nevertheless, the acoustic idiosyncrasies of a specific training domain and knowledge about its environmental conditions are two features which are implicitly considered by the algorithm presented in this paper, since the type of acoustic models used intrinsically influence the solution. “Human knowledge” as we broadly allude to here does not generally include these two sources. The algorithm in this paper thus presents a method to take these into account while designing a phoneset and dictionary for a particular task.

Appendix A
Iterative Procedure for Joint Optimization of Two Variables

In the first part of this Appendix we derive an iterative procedure for the joint MAP estimation of two random variables. The second part derives a similar procedure for the joint estimation of two random variables where *a priori* constraints exist for only one of the two variables.

Problem A: Find $\hat{a}$ and $\hat{b}$ such that

$$\hat{a},\, \hat{b} = \arg\max_{a,\, b} P(a,\, b|c). \tag{29}$$

Let the ith estimate of $\hat{a}$ and $\hat{b}$ be a_i and b_i, respectively. Let

$$a_{i+1} = \arg\max_{a} P(a|b_i,\, c). \tag{30}$$

It is easy to show using Bayes’ rule that

$$a_{i+1} = \arg\max_{a} P(a,\, b_i|c). \tag{31}$$

Therefore

$$P(a_{i+1},\, b_i|c) \geq P(a_i,\, b_i|c). \tag{32}$$

Similarly, if

$$b_{i+1} = \arg\max_{b} P(b|a_{i+1},\, c) \tag{33}$$

we get

$$P(a_{i+1},\, b_{i+1}|c) \geq P(a_{i+1},\, b_i|c). \tag{34}$$

Therefore, iterations of (30) and (33) result in increasing values of $P(a,\, b|c)$, leading to a locally optimal estimate of $\hat{a}$ and $\hat{b}$.

Problem B: Find $\hat{a}$ and $\hat{c}$ such that

$$\hat{a},\, \hat{c} = \arg\max_{a,\, c} P(a,\, b|c). \tag{35}$$

Using logic very similar to that used for $\arg\max_{a,\, b} P(a,\, b|c)$, it can be shown that a locally optimal estimate of $\hat{a}$ and $\hat{c}$ can be obtained by iterations of

$$c_i = \arg\max_{c} P(b|a_i,\, c)P(a_i|c) \tag{36}$$

$$a_{i+1} = \arg\max_{a} P(a|b,\, c_i). \tag{37}$$

Appendix B
Maximum a Posteriori Estimation of Multiple Pronunciations for a Word

In this Appendix, we briefly outline a procedure with which the approach discussed in the body on of this paper can be extended to accommodate multiple pronunciations for any given word.

For simplicity, let us assume that the word has only two pronunciations. Let A_w represent the set of acoustic data from all instances of the word w. The *a posteriori* probability of any set of two phone sequences $\{\wp_1,\, \wp_2\}$, conditioned on A_w is given by

$$\begin{aligned} &P(\{\wp_1,\, \wp_2\}|A_w,\, \lambda_n,\, E_{spel},\, E_{seq}) \\ &\quad = \frac{P(\{\wp_1, \wp_2\}|E_{spel}, E_{seq})P(A_w|\{\wp_1, \wp_2\}, \lambda_n, E_{spel}, E_{seq})}{P(A_w|\lambda_n, E_{spel}, E_{seq})}. \end{aligned} \tag{38}$$

Here, and in the rest of this Appendix, we have assumed that the phone sequences are independent of the acoustic model λ_n when the two are not related by acoustic data. We note that the denominator in (38) is not a function of the phone sequences $\wp_1$ and $\wp_2$. As in the rest of the paper, we also assume that the likelihood of the acoustic data is independent of spelling and phone-sequence constraints, E_{spel} and E_{seq}, once the specific pronunciations for the word are given.

Let A_{w_k} represent the acoustic data from the kth example of w in A_w. We assume that the various instances of the word are independent of each other. Thus,

$$\begin{aligned} &P(A_w|\{\wp_1,\, \wp_2\},\, \lambda_n,\, E_{spel},\, E_{seq}) \\ &\qquad = \prod_k P(A_{w_k}|\{\wp_1,\, \wp_2\},\, \lambda_n). \end{aligned} \tag{39}$$

The likelihood of any A_{w_k} is given by

$$\begin{aligned} &P(A_{w_k}|\{\wp_1,\, \wp_2\},\, \lambda_n) \\ &\quad = P(A_{w_k},\, \wp_1|\{\wp_1,\, \wp_2\},\, \lambda_n) + P(A_{w_k},\, \wp_2|\{\wp_1,\, \wp_2\},\, \lambda_n) \end{aligned} \tag{40}$$

$$\quad = P_{\wp_1}P(A_{w_k}|\wp_1,\, \lambda_n) + P_{\wp_2}P(A_{w_k}|\wp_2,\, \lambda_n) \tag{41}$$

where $P_{\wp_1}$ and $P_{\wp_2}$ are the *a priori* probabilities for the two pronunciations $\wp_1$ and $\wp_2$ for the word w (assuming that there are only two pronunciations $\wp_1$ and $\wp_2$ for the word). Representing the two pronunciations of the word w as $\wp_{w_1}$ and $\wp_{w_2}$, respectively, and combining (38), (39), and (41), we get the *maximum a posteriori* estimate of $\wp_{w_1}$ and $\wp_{w_2}$ as

$$\begin{aligned} \{\wp_{w_1},\, \wp_{w_2}\} = &\arg\max_{\wp_1,\, \wp_2}\{P(\{\wp_1,\, \wp_2\}|E_{spel},\, E_{seq}) \\ &\cdot \prod_k \{P_{\wp_1}P(A_{w_k}|\wp_1,\, \lambda_n) + P_{\wp_2}P(A_{w_k}|\wp_2,\, \lambda_n)\}\}. \end{aligned} \tag{42}$$

Thus, the *maximum a posteriori* estimate for the two pronunciations of w is obtained by computing the argument in (42) for every pair of phone sequences and identifying the pair for which it is maximum. Within any pair of pronunciations, $P_{\wp_1}$ and $P_{\wp_2}$ would have to be computed as the expected fraction of examples of the word that get classified as belonging to $\wp_1$

and $\wp_2$ respectively. Alternately, $P_{\wp_1}$ and $P_{\wp_2}$ could be directly computed from E_{spel} and E_{seq} by invalidating the assumption that the likelihood of the acoustic data is independent of spelling and phone constraints when the pronunciations for the word are given. If the number of possible pronunciations can be constrained in any manner to a small set, exhaustive evaluation of (42) may be possible. Otherwise, locally optimal iterative solutions may be required.

It is easy to generalize the above formulation for any specific number of pronunciations. However, the determination of the exact number of pronunciations for a word would require evaluation of (42) for all possible numbers of pronunciations and validation on a held out set.

References

[1] T. Svendsen, K. K. Paliwal, E. Harborg, and P. O. Husoy, "An improved subword based speech recognizer," in *Proc. IEEE Conf. Acoustics, Speech, Signal Processing*, 1989, pp. 108–111.

[2] T. Svendsen, F. K. Soong, and H. Purnhagen, "Optimizing baseforms for HMM-based speech recognition," in *Proc. Eur. Conf. Speech Communication Technology (EUROSPEECH)*, 1995, pp. 783–786.

[3] L. R. Bahl, P. F. Brown, P. V. de Souza, R. L. Mercer, and M. A. Picheny, "A method for the construction of acoustic Markov models for words," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 443–452, Oct. 1993.

[4] T. Holter and T. Svendsen, "Combined optimization of baseforms and model parameters in speech recognition based on acoustic subword units," in *Proc. IEEE Workshop Automatic Speech Recognition*, 1997, pp. 199–206.

[5] M. Bacchiani and M. Ostendorf, "Joint lexicon, acoustic unit inventory and model design," *Speech Commun.*, vol. 29, pp. 99–114, 1999.

[6] J. G. Wilpon, B. H. Juang, and L. R. Rabiner, "An investigation on the use of acoustic sub-word units for automatic speech recognition," in *Proc. IEEE Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 1987, pp. 821–824.

[7] T. Sloboda and A. Waibel, "Dictionary learning for spontaneous speech recognition," in *Proc. Int. Conf. Speech Language Processing (ICSLP)*, 1996, pp. 2328–2331.

[8] M. Ravishankar and M. Eskenazi, "Automatic generation of context-dependent pronunciations," in *Proc. Eur. Conf. Speech Communication and Technology (EUROSPEECH)*, 1997, pp. 2467–2470.

[9] M. B. Wesenick, "Automatic generation of German pronunciation variants," in *Proc. Int. Conf. Speech Language Processing (ICSLP)*, 1996, pp. 125–128.

[10] T. Holter and T. Svendsen, "Incorporation of linguistic knowledge and automatic baseform generation in acoustic subword unit based speech recognition," in *Proc. Eur. Conf. Speech Communication Technology (EUROSPEECH)*, 1997, pp. 1159–1162.

[11] A. P. Dempster, N. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc.*, vol. B39, pp. 1–38, 1977.

[12] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[13] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Reading, MA: Addison-Wesley, 1979.

[14] S. Porat and J. Feldman, "Learning automata from ordered examples," *Mach. Learn.*, vol. 7, pp. 109–138, 1991.

[15] N. J. Nilsson, *Problem Solving Methods in Artificial Intelligence*. New York: McGraw-Hill, 1971.

[16] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 400–401, Mar. 1987.

[17] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer, "The mathematics of statistical machine translation," *Comput. Linguist.*, vol. 19, pp. 263–311, 1993.

[18] ——, "Automatic clustering and generation of contextual questions for tied states in hidden Markov models," in *Proc. IEEE Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 1999, pp. 117–120.

[19] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallet, "The DARPA 1000-Word Resource Management database for continuous speech recognition," in *Proc. IEEE Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 1988, pp. 651–654.

[20] X. D. Huang, K. F. Lee, H. W. Hon, and M. Y. Hwang, "Improved acoustic modeling with the SPHINX speech recognition system," in *Proc. IEEE Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 1991, pp. 345–348.

[21] W. M. Fisher, "A statistical text-to-phone function using ngrams and rules," in *Proc. IEEE Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 1999, pp. 649–652.

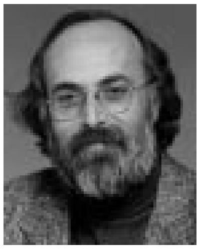
[22] Carnegie Mellon Univ., Pittsburgh, PA. [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

Rita Singh received the B.Sc.(Hons.) degree in physics and the M.Sc. degree in exploration geophysics, both from the Banaras Hindu University, India. She received the Ph.D degree in geophysics in 1996 from the National Geophysical Research Institute of the Council of Scientific and Industrial Research, India.

She is a Member of the Research Faculty at the School of Computer Science, Carnegie Mellon University (CMU), Pittsburgh, PA, and a Visiting Scientist with the Media Labs, Massachusetts Institute of Technology, Cambridge. From March 1996 to November 1997, she was a Postdoctoral Fellow with the Tata Institute of Fundamental Research, India, where she worked with the Condensed Matter Physics and Computer Systems and Communications Groups. During this period, she worked on nonlinear dynamical systems and signal processing as an extension of her doctoral work on nonlinear geodynamics and chaos. Since November 1997, she has been affiliated with the Robust Speech Recognition and SPHINX Groups at CMU, and has been working on various aspects of speech recognition including core HMM-based recognition technology, automatic learning techniques, and environmental robustness techniques for speech recognition.

Bhiksha Raj received the M.Tech degree in electronics and communication engineering from the Indian Institute of Technology, Madras, and the Ph.D degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2000. His doctoral work was in the area of missing feature methods for robust speech recognition.

He is a Research Scientist with Mitsubishi Electric Research Laboratories, Cambridge, MA. From 1991 to 1994, he was with the Computer Systems and Communication Group at the Tata Institute of Fundamental Research, Bombay, India. From 2000 to 2001, he was with Compaq Computer Corporation, where he worked on multiple-microphone based approaches for robust speech recognition, and language model compression. His current research interests include multiple-microphone-based speech processing, distributed speech recognition, and automatic learning techniques for integration of multiple information sources for speech recognition.



Richard M. Stern (M'77) received the S.B. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 1970, the M.S. degree from the University of California, Berkeley, in 1972, and the Ph.D. degree from MIT in 1977, all in electrical engineering.

He has been on the faculty of Carnegie Mellon University (CMU), Pittsburgh, PA, since 1977, where he is currently a Professor of electrical engineering and computer science, and Associate Director of the CMU Information Networking Institute. Much of his current research is in spoken language systems, where he is particularly concerned with the development of techniques with which automatic speech recognition can be made more robust with respect to changes in environment and acoustical ambience. He has also developed sentence parsing and speaker adaptation algorithms for earlier CMU speech systems. In addition to his work in speech recognition, he also maintains an active research program in psychoacoustics, where he is best known for theoretical work in binaural perception.

Dr. Stern was a co-recipient of CMU's Allen Newell Medal for Research Excellence in 1992. He is a member of the Acoustical Society of America.