BAUM-WELCH TRAINING FOR SEGMENT-BASED SPEECH RECOGNITION

Han Shu, I. Lee Hetherington, and James Glass

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{hshu, ilh, jrg}@sls.lcs.mit.edu

ABSTRACT

The use of segment-based features and segmentation networks in a segment-based speech recognizer complicates the probabilistic modeling because it alters the sample space of all possible segmentation paths and the feature observation space. This paper describes a novel Baum-Welch training algorithm for segment-based speech recognition which addresses these issues by an innovative use of finite-state transducers. This procedure has the desirable property of not requiring initial seed models that were needed by the Viterbi training procedure we have used previously. On the PhoneBook telephone-based corpus of read, isolated words, the Baum-Welch training algorithm obtained a relative error reduction of 37% on the training set and a relative error reduction of 5% on the test set, compared to Viterbi trained models. When combined with a duration model, and more flexible segmentation network, the Baum-Welch trained models obtain an overall word error rate of 7.6%, which is the best result we have seen published for the 8,000 word task.

1. INTRODUCTION

The use of mathematically rigorous hidden Markov models (HMMs) has in part contributed to the dramatic improvement in automatic speech recognition (ASR) over the last two decades. The acoustic models in HMM ASR systems model a temporal sequence of feature vectors computed at a fixed frame-rate, most commonly at 10ms/frame. Since the duration of a typical phone can vary from 20ms to over 200ms, the number of fixed frame-rate feature vectors within the same phonetic segment is usually much greater than one. These feature vectors within the same phonetic segment are typically highly correlated. However, HMMs have an inherent conditional independence assumption on the observation feature vectors. Thus, the fixed frame-rate feature vector employed by HMM-based recognizers fundamentally limits the range of acoustic models that can be explored for encoding acoustic-phonetic information. While many research groups have focused on improving frame-based HMM ASR systems, some groups have tried to avoid this limitation by constructing segment-based ASR systems [3, 5, 10].

The acoustic models in a segment-based ASR system model a sequence of feature vectors computed on time intervals that are not necessarily equal. The segment-based ASR system developed in our group, the SUMMIT system, uses two different types of feature vectors, namely *segment features* and *landmark features* [6]. The segment features are computed from the portion of the speech waveform belonging to a hypothesized phonetic segment, and the landmark features are computed from fixed-size waveform intervals centered at landmarks. The landmark feature framework is motivated by the belief that acoustic cues important for phonetic classification are located at acoustic landmarks corresponding to oral closure (or release) or other points of maximal constriction (or opening) in the vocal tract [13]. The segment feature framework promotes flexible modeling of phonetic segments without the conditional independence assumption imposed by HMMs. In SUMMIT, the segment features and landmark features can be used jointly or separately.

The SUMMIT segment-based recognizer consists of two major components. The first component proposes segments, and the second models the acoustic observations on the segments. A segment-based ASR system either implicitly or explicitly hypothesizes segmentations of the speech waveform, although SUMMIT typically uses explicit segmentation, especially for real-time performance. It is worth noting that the first component does not simply hypothesize a single sequence of non-overlapping segments; rather it produces a segment network, which allows a set of segmentation sequences to be encoded. The use of a segment network reduces the accuracy requirement on the first component, thus increasing the robustness of the overall segment-based system. Frame-based HMM ASR systems do not generate a

This research was supported by DARPA under contract N66001-99-1-8904 monitored through Naval Command, Control and Ocean Surveillance Center, by the NSF under award 9872995-1483, and by the Spoken Language Systems Group Affiliates Program.

segment network. The frame-based approach can be viewed as using an implicit fully-connected segment network.

The SUMMIT recognizer also deploys a probabilistic decoding strategy. For conventional speech recognizers, the Baum-Welch training algorithm has been shown to have a smoother convergence property than the Viterbi training, currently used by some segment-based systems. The use of segment-based features and segmentation networks complicates the probabilistic modeling because it alters the sample space of all possible segmentation paths and the feature observation space. Viterbi-based training avoids these complications by only learning from the single best forced alignment for a given initial model. This paper describes a novel Baum-Welch training algorithm for segment-based speech recognition, which addresses these complications by an innovative usage of the finite state transducer. It is important to note that Baum-Welch training was used for the segment-based recognition systems in [3, 10]; however these systems do not have the same difficulties from their feature vectors and segmentation network. In these studies the feature vectors are uniformly sampled, as in a typical frame-based recognition system. The segmentation networks are also similar to those of a frame-based system, an implicit fully-connected segment network.

In the following sections we first describe the probabilistic formulation used for segment-based ASR, and then describe the Baum-Welch training procedure we have developed that accounts for the constrained segmental search space. We then report experimental results obtained on the PhoneBook telephone-based corpus of read, isolated words, where we compare the Baum-Welch training against the Viterbi training procedure we have used previously. Finally, we discuss benefits and trade-offs between Viterbi training and Baum-Welch training for segment-based ASR and describe our future plans for improving both segment-based and frame-based recognition.

2. PROBABILISTIC FOUNDATION OF SEGMENT-BASED ASR

In the typical formulation, the goal of recognition is to find the sequence of words $\vec{W}^* = W_1, \ldots, W_N$ which gives the maximum a posteriori probability given the acoustic observations $\vec{O}$, that is:

$$\vec{W}^* = \arg\max_{\vec{W}} P(\vec{W}|\vec{O}) = \arg\max_{\vec{W}} P(\vec{W}, \vec{O}), \tag{1}$$

where $\vec{W}$ ranges over all possible word sequences. In most ASR systems, a sequence of sub-word units, $\vec{U}$, and a sequence of sub-phone states, $\vec{S}$, are decoded along with the optimal word sequence. Eq. 1 becomes:

$$\begin{aligned} \vec{W}^* &= \arg\max_{\vec{W}} \sum_{\forall \vec{S}, \vec{U}} P(\vec{S}, \vec{U}, \vec{W}, \vec{O}) \\ &\approx \arg\max_{\vec{S}, \vec{U}, \vec{W}} P(\vec{S}, \vec{U}, \vec{W}, \vec{O}). \end{aligned} \tag{2}$$

The approximation in Eq. 2 is commonly known as the "Viterbi approximation." The expression $P(\vec{S}, \vec{U}, \vec{W}, \vec{O})$ can be decomposed into the form:

$$\begin{aligned} &P(\vec{S}, \vec{U}, \vec{W}, \vec{O}) \\ &\quad = P(\vec{O}|\vec{S}, \vec{U}, \vec{W}) P(\vec{S}|\vec{U}, \vec{W}) P(\vec{U}|\vec{W}) P(\vec{W}). \end{aligned} \tag{3}$$

With appropriate conditional independence assumptions, the term $P(\vec{S}, \vec{U}, \vec{W}, \vec{O})$ becomes,

$$P(\vec{S}, \vec{U}, \vec{W}, \vec{O}) = P(\vec{O}|\vec{S}) P(\vec{S}|\vec{U}) P(\vec{U}|\vec{W}) P(\vec{W}). \tag{4}$$

$P(\vec{O}|\vec{S})$ is the usual acoustic model. The term $P(\vec{S}|\vec{U})$ is the weighted mapping between the sequences of sub-word units to sequences of sub-phone units. The term $P(\vec{U}|\vec{W})$ describes the sequences of sub-word units that can be generated for a given word sequence, typically accomplished by a dictionary lookup table and phonological rules to model systematic phonological variations in fluent speech. $P(\vec{W})$ is the language model.

In the SUMMIT segment-based speech recognition system [17], various constraints such as the acoustic model, A, model topology, M, context dependency, C, phonological rules [8], P, lexicon, L, and language model, W, are all represented by weighted finite-state transducers (FSTs). With these FSTs, the joint probability in the right hand side of Eq. 4 has an FST equivalent,

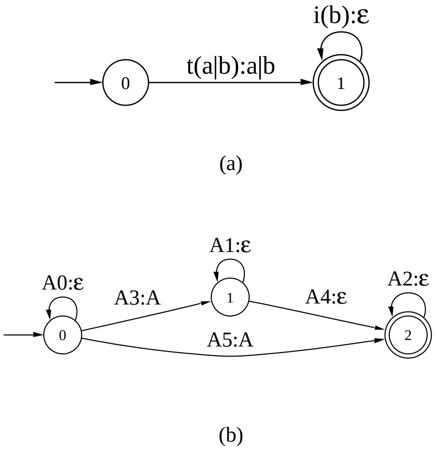
$$\begin{array}{ccccccc} \underbrace{P(\vec{O}|\vec{S})} & \cdot & \underbrace{P(\vec{S}|\vec{U})} & \cdot & \underbrace{P(\vec{U}|\vec{W})} & \cdot & \underbrace{P(\vec{W})} \\ \Big\downarrow & & \Big\downarrow & & \Big\downarrow & & \Big\downarrow \\ A & \circ & M & \circ & (C \circ P \circ L) & \circ & G \end{array} \tag{5}$$

The recognition problem of Eq. 2 is thus converted to the equivalent problem of searching for the best path in $A \circ M \circ C \circ P \circ L \circ G$.

A natural question to ask is "where is the segment network constraint in Eq. 5?" It is actually hidden inside the first FST A. In this case, the sequences of sub-phone states, $\vec{S}$, contain phonetic or even syllabic landmarks. The set of mappings between sequences of observation vectors and the sub-phone state sequences encoded in A is limited by the segment network. With the segment network constraint, the FST A is less "bushy" than without. The FST A can be thought of as the composition of two FSTs, $A_S \circ A_M$, where the FST A_S represents the segment network constraint with the output symbol "#p" for marking phonetic boundaries, and the FST A_M simply translates the output symbol "M" into the set of all possible sub-phone states. Figure 2 shows a sample segment network, and its corresponding FST representations for landmark features, A_S.

2.1. Landmark Models

The segment-based landmark models in SUMMIT are a generalized version of those in a frame-based HMM ASR system. The two systems differ in three aspects. First, the observation feature vector for landmark models is not limited to a fixed frame-rate feature vector, but is rather sampled non-uniformly. Whether uniformly sampled or not, it is important to note that in both systems all the input sequences are the same on different segmentation paths. Second, the segment network in segment-based systems constrains the search space, whereas HMM-based system do not. The segment network constraints can be relaxed to produce a fully-connected network like the one used by HMMs. Third, the model topology FST M currently used by SUMMIT is different from that of an HMM, as illustrated in Figure 1. In summary, the segment-based SUMMIT ASR system implemented with FSTs is a very flexible framework. It can be easily configured to implement an HMM by appropriately altering the FSTs A_S and M, and the observation feature vectors $\vec{O}$.



(a)

(b)

Fig. 1. Illustration of the model topology FSTs M. (a) is used by the current SUMMIT landmark features, and (b) is for a 3-state HMM with skip transitions.

3. BAUM-WELCH TRAINING OF SEGMENT-BASED ACOUSTIC MODELS

Currently, the segment-based acoustic models in SUMMIT are trained with a procedure called segmental K-means, or Viterbi training [12, 6]. In Viterbi training, each observation is assigned to a *single* acoustic model. For most HMM-based speech systems, the acoustic models are trained with Baum-Welch training, in which each observation is assigned to a *set* of acoustic models with weights [12]. Only a portion of each observation, equal to its posterior probability, is associated with each model. Many studies have found that for HMM-based systems, the Baum-Welch trained acoustic models outperform Viterbi-trained ones. However, it is not known whether Baum-Welch training of segment-based acoustic models would improve recognition performance.

The newly proposed Baum-Welch training of segment-based acoustic models consists of two steps. First, the "expectation" step (or E step) computes the posterior probabilities, $\gamma_n(i)$ defined as:

$$\gamma_n(i) = P(q_n = i|\vec{O}, \lambda) \quad \forall i = 1, 2, \ldots, K, \tag{6}$$

where the random variable q_n is equal to integer i when the observation O_n belongs to the i^{th} acoustic model, $\vec{O}$ is a sequence of N observations, $\{O_1, O_2, \ldots, O_N\}$, λ is the parameter set for the current acoustic models, and K is the number of acoustic models. The posterior, $\gamma_n(i)$, is the probability that the n^{th} observation belongs to the i^{th} acoustic model. The acoustic model in this case is the landmark model. Second, the "maximization" step (or M step) trains observation probability density functions (PDFs) with the posterior-weighted observations for every acoustic model. In the following sections, we will describe the details of these two steps.

3.1. Computation of the Posterior Probabilities

To compute the posterior probabilities, we employ the standard equation using the forward probability, $\alpha_n(i)$, and backward probability, $\beta_n(i)$,

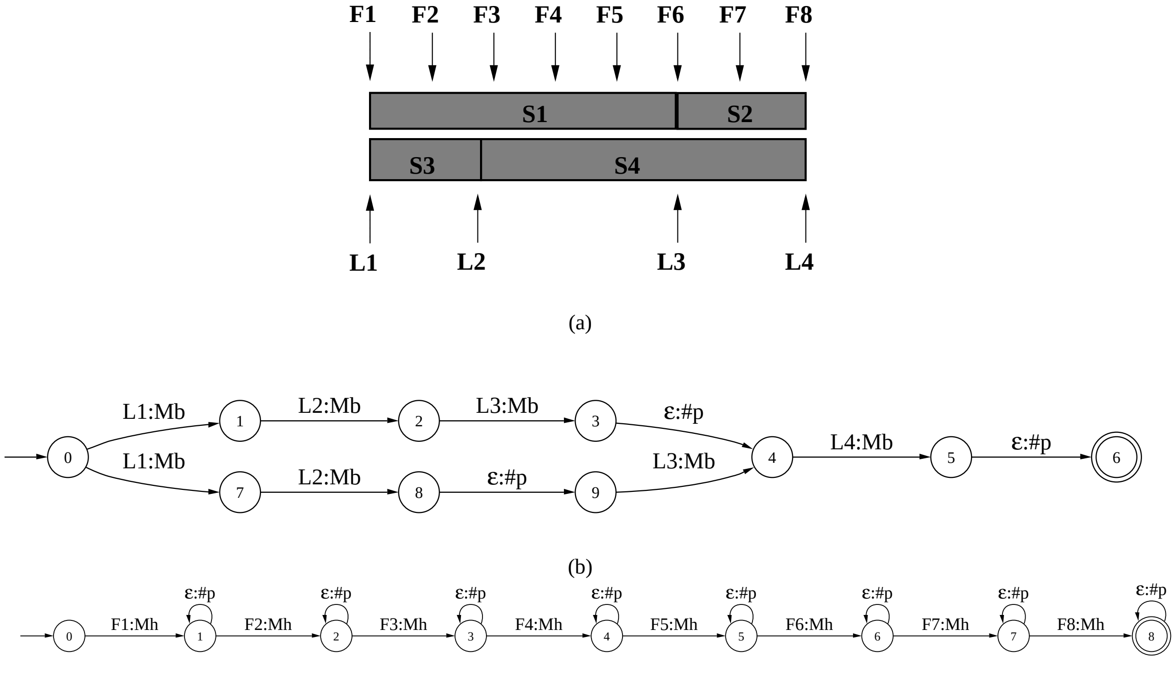
$$\gamma_n(i) = \frac{\alpha_n(i)\beta_n(i)}{\sum_{i=1}^{K} \alpha_n(i)\beta_n(i)}, \tag{7}$$

where $\alpha_n(i)$ and $\beta_n(i)$ are defined as,

$$\alpha_n(i) = P(O_1 O_2 \ldots O_n, q_n = i|\lambda), \tag{8}$$

$$\beta_n(i) = P(O_{n+1} O_{n+2} \ldots O_N|q_n = i, \lambda). \tag{9}$$

In HMM-based ASR systems, there is no segment network which constrains the mapping between feature observations and acoustic models. However, in a segment-based ASR system, the segment network *does* constrain the possible mappings between observations and acoustic models. This segment network constraint needs to be taken into account when computing the $\alpha_n(i)$ and $\beta_n(i)$ variables. This is the key difference between Baum-Welch training for HMM models and segment-based models.



(a)

(b)

(c)

Fig. 2. Illustration of a sample segment network and its corresponding FST representation. Here only the FST A_S is shown since FST A_M simply translates the input symbol Mb, Ms, Mh into the set of all possible sub-phone states. The segment network in (a) contains four phonetic segments with four landmark feature vectors, $L1$, $L2$, $L3$, and $L4$, and four segment feature vectors, $S1$, $S2$, $S3$, and $S4$. The feature vectors, $F1$, $F2$, $\ldots$, $F8$ are the corresponding fixed frame-rate feature vectors using by HMMs. (b) shows the corresponding FST A_S for landmark features with two identical input sequences, $L1L2L3L4$, and the symbol Mb represents the set of all landmark models. The symbol $\#p$ denotes phone landmark locations. (c) shows the corresponding FST A_S for a frame-based HMM. Since the symbol $\#p$ in (c) does not provide any constraint, the size of the corresponding $A = A_S \circ A_M$ is typically bigger than that of segment-based models in (b).

Given a sequence of observations, $\vec{O}$, and its corresponding segment network, S, one can construct an FST, A, that specifies all possible mappings between each observation, O_i, and each state variable q_n. This is done in two steps. We first convert the segment network, S into its FST representation, A_S, then FST A_S is composed with FST A_M to form FST A. Let W be the linear FST representing the sequence of reference words, $\vec{W}$. An FST, Z, conforming to the segment network A and reference word sequence $\vec{W}$ can be computed by a sequence of FST operations, namely,

$$Z = A \circ project_I(M \circ C \circ P \circ L \circ W). \qquad (10)$$

The constraint lattice represented by FST Z encodes all possible mappings between O_i and q_n given the segment network and reference word sequence.

As described in Sec. 2, FSTs C, P, and L represent various other constraints, and the FST M represents the model topology used by the recognizer. When the FST Z is computed for each tuple $\{S, \vec{O}, \vec{W}\}$, the forward and backward variables $\alpha_n(i)$ and $\beta_n(i)$ can be computed on the network specified by Z. Finally, $\gamma_n(i)$ can be computed from $\alpha_n(i)$ and $\beta_n(i)$ according to Eq. 7. The second term on the right-hand side of Eq. 10 is an acceptor for the (possibly infinite) sequences of sub-phone units implied by the word sequence, $\vec{W}$. They are then mapped to acoustic observations by FST A.

3.2. Train Observation PDFs from Posterior-Weighted Feature Vectors

The observation PDFs for acoustic models are typically in the form of Gaussian mixture models (GMMs), because of their modeling power and their computational efficiency. The current SUMMIT implementation already uses the EM training of Gaussian mixture models from feature vectors with unity weights [1, 2]. The EM training of the Gaussian components can be done via the "split and merge" procedure [16], "k-means" [4], or "model aggregation" [7]. Since the first step of k-means is a random initialization of the centroids, the resulting Gaussian mixture models can vary in performance from different initializations. Experimentally the split and merge procedure matches the best performance of multiple training runs with different k-means initializations. We have observed consistent WER improvement from using the model aggregation. For this work, only the split and merge procedure is used. We will explore using model aggregation in the future.

To train GMMs from posterior-weighted feature vectors instead of unity weighted ones, the training procedure needs to be modified slightly. To complete a Baum-Welch training iteration, the update equations needs to simply take the posterior probabilities $\gamma_n(i)$ weighting into account.

4. EXPERIMENT & DISCUSSION

We have experimented the new Baum-Welch training on landmark feature observations for the PhoneBook task [11]. The PhoneBook telephone-based corpus consists of read, isolated words from a vocabulary of close to 8,000 words. In the baseline systems the landmark models were Viterbi trained [9]. As defined in [9], we focus on the harder task of the "large" set containing about 80,000 training utterances and 7,000 test sentences, with a decoding vocabulary of 8,000 words.

The baseline word error rate (WER) on the training is 4.3%, and on the test is 9.9%. This baseline is with landmark acoustic models only. In [9] Livescu et al. also presented a WER of 8.7% with duration models. Since we are focused on Baum-Welch training of the landmark models in this paper, we only compare it with the results of landmark models. Table 1 summarizes the results of WERs of the baseline systems and of Baum-Welch trained models. The Baum-Welch trained acoustic models achieved a relative error reductions of 37% on training, and a relative error reductions of 5% on test. The WER improved significantly on training, but on test the improvement was much smaller.

Training Method	# Params	Training WER	Test WER
Viterbi	1.55M	4.3%	9.9%
Baum-Welch	*1.64M*	*2.7%*	*9.4%*

Table 1. Word error rates (WER) of segment-based recognizer training using Viterbi training and Baum-Welch training on the training set and test set.

Although the WER improvement on the test is small, Baum-Welch training has a desirable advantage over Viterbi training. Viterbi training requires an initial set of acoustic models for forced alignment of the training data, whereas Baum-Welch training is bootstrapped with flat initialization models—mixtures with single zero-mean unit-variance Gaussian components. The performance of Viterbi trained acoustic models is thus dependent on the quality of the initial models. Since the initial models are typically learned from additional data, the implicit training set is arguably bigger than the stated training set. More seriously however, in some cases the initialization required by Viterbi training is difficult to obtain. For example, when Tang et al. experimented with a two stage recognition system in which the first stage is a recognizer using a reduced phone set [15], the requirement of good initialization models limits the types of reduced phone sets to be a many-to-one mapping of an existing recognizer's phone set. Because Baum-Welch training does not require any pre-trained initial acoustic model, the set of reduced phone set are not limited. However, Baum-Welch training is slower since it has to iterate through the training data a number of times. On the PhoneBook task, Baum-Welch training is about ten times slower than the Viterbi training baseline.

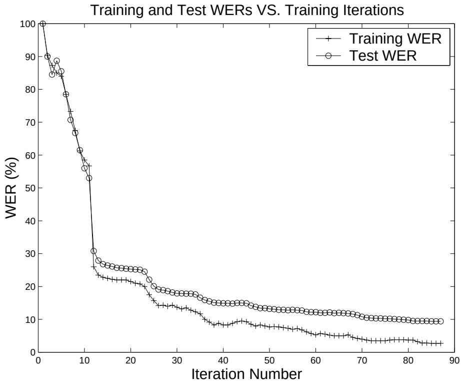


Fig. 3. Training and test WERs as a function of training iterations. The upper curve is the test WERs, and the lower curve is the training WERs. As the training iteration increases, the number of parameters in the acoustic models also increases. The WERs of 100.0% from the first iteration is from the flat initialization models. After a total of 87 iterations, the training WER converges to 2.7%, and the test WER converges to 9.4%.

5. FUTURE

The work reported in this paper summarizes our initial efforts in converting the training process of our segment-based speech recognizer to Baum-Welch training. Our initial efforts focused on converting the landmark model training. Previous works have shown improved WER performance with the combination of landmark models and segment models [14] and the combination of landmark models and duration models [9]. Since these models were all Viterbi trained, we are optimistic that similar improvements will be achieved with Baum-Welch trained models. We therefore plan to extend the Baum-Welch training to the segment models and the duration models. Similar constraint lattices represented by FST Z can be computed for segment and duration features. We have worked out these problems mathematically, and are currently implementing them.

In addition to converting our training procedure to Baum-Welch, we are also exploring the effect of varying the size of our segment network, since the effect of the segmentation network on the overall recognition system performance is not well understood. For example, with a less constrained segment network, and Viterbi trained duration models, we achieve a PhoneBook test WER of 7.6%, which we believe is the lowest reported result on this task.

Finally, we are ultimately interested in exploring the benefits of combining frame-based and segment-based acoustic modeling. We are currently modifying our recognizer so it can accommodate a more complicated model topology, and so it can decode without a segmentation network. With the completion of these modifications, the SUMMIT recognizer will have a common framework for both frame-based and segment-based recognition. The common framework will enable us to ultimately compare and combine the frame-based and segment-based systems so that we can investigate the fusion of the frame-based and the segment-based approaches without lattice re-scoring.

6. ACKNOWLEDGMENTS

We would like to thank Karen Livescu for her help in providing the baseline PhoneBook recognizer and answering many PhoneBook-related questions.

7. REFERENCES

[1] J. Bilmes, “A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models,” Tech. Rep. ICSI-TR-97-021, University of Berkeley, 1997.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, Jun. 1977.

[3] V. V. Digilakis, *Segment-based stochastic models of spectral dynamics for continuous speech recognition*. Ph.D. thesis, Boston University, Jan. 1992.

[4] R. Duda and P. Hart, *Pattern classification and scene analysis*. New York, Chichester, Brisbane, Toronto, Singapore: John Wiley & Sons, 1973.

[5] H. Gish and K. Ng, “Parametric trajectory models for speech recognition,” in *Proc. Intl. Conf. on Spoken Language Processing*, Philadelphia, PA, vol. 1, pp. 466–469, Oct. 1996.

[6] J. R. Glass, “A probabilistic framework for segment-based speech recognition,” *Computer Speech and Language*, vol. 17, no. 2-3, pp. 137–152, 2003.

[7] T.J. Hazen and A.K. Halberstadt, “Using aggregation to improve the performance of mixture Gaussian acoustic models,” in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Seattle, WA, pp. 653–656, May 1998.

[8] I. L. Hetherington, “An efficient implementation of phonological rules using finite-state transducers,” in *Proc. European Conf. on Speech Communication and Technology*, Aalborg, pp. 1599–1602, Sept. 2001.

[9] K. Livescu and J. Glass, “Segment-based recognition on the PhoneBook task: initial results and observations on duration modeling,” in *Proc. European Conf. on Speech Communication and Technology*, Aalborg, Denmark, pp. 1437–1440, Sept. 2001.

[10] M. Ostendorf, V. Digilakis, and O. Kimball, “From HMM’s to segment models: a unified view of stochastic modeling for speech recognition,” *IEEE Trans. Speech and Audio Processing*, vol. 4, no. 5, pp. 360–378, 1996.

[11] J. Pitrelli, C. Fong, S. Wong, J. Spitz, and H. Leung, “PhoneBook: A phonetically-rich isolated-word telephone-speech database,” in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, Michigan, vol. 1, pp. 101–104, May 1995.

[12] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, pp. 257–286, 1989.

[13] K. Stevens, “Applying phonetic knowledge to lexical access,” in *Proc. European Conf. on Speech Communication and Technology*, Madrid, Spain, pp. 3–11, Sept. 1995.

[14] N. Ström and I. L. Hetherington and T. J. Hazen and E. Sandness and J. R. Glass, “Acoustic modeling improvements in a segment-based speech recognizer,” in *IEEE Automatic Speech Recognition and Understanding Workshop*, Snowbird, pp. 139–142, Dec. 1999.

[15] M. Tang, S. Seneff, and V. W. Zue, “Modeling linguistic features in speech recognition,” in *Proc. European Conf. on Speech Communication and Technology*, Geneva, Switzerland, Sept. 2003.

[16] S. Young, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK book*, Cambridge, UK: Cambridge University, 1997.

[17] V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, and I. L. Hetherington, “JUPITER: A telephone-based conversational interface for weather information,” *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 1, pp. 100–112, 2000.