Recent Progress on the SUMMIT System

Victor Zue, James Glass, David Goodine, Hong Leung,
Michael Phillips, Joseph Polifroni, and Stephanie Seneff
Room NE43-601
Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

Introduction

The SUMMIT system is a speaker-independent, continuous-speech recognition system that we have developed at MIT [12]. To date, the system has been ported to a variety of tasks with vocabulary sizes up to 1000 words and perplexities up to 73. The architecture of this system is a product of two guiding principles. First, we desired a framework that could be flexible and modular so that we could explore alternative strategies for embedding speech knowledge into the system. Second, we required that the system be stochastic and trainable from a large body of speech data to account for our current incomplete knowledge of the acoustic realization of speech. The current implementation of the system is a reflection of both of these ideas. SUMMIT differs from the majority of prevailing HMM approaches in many respects ranging from its use of auditory models and selected acoustic measurements, to its segmental framework and use of pronunciation networks. In time, the specific implementation of these ideas will undoubtedly be modified as we discover superior techniques and approaches. Until phonetic and word recognition accuracies are competitive with those of human listeners however, we believe it will be appropriate to incorporate both notions of flexibility and trainability into the system.

In the past year we have focused our attention on a larger spoken-language effort which integrates SUMMIT with a natural language system. We have also investigated issues which relate to phonetic recognition; namely alternative segmental representations and classification techniques. In addition, we have changed our normalization procedure to make it more amenable for recording spontaneous speech. In this paper we first describe the normalization procedure. We then review our segmental framework, and describe two alternatives we investigated. Finally, we present some phonetic classification and recognition experiments which assess the different segmental representations and classification techniques. These experiments indicated an improvement in classification rate of 4%, and in recognition rate of 8%.

Normalization

Most speech recognition systems that measure some type of absolute value, such as the short-term energy, require that an utterance is first normalized before processed. In the SUMMIT system for instance, a weak signal that is not normalized will produce auditory outputs which are smaller than the spontaneous firing rate and will therefore be indistinguishable from silence. Currently we use a normalization procedure that scales a speech waveform with respect to its maximum magnitude value. This simple procedure has proven quite effective for the majority of speech processing and recognition applications since 1) the utterance has typically been completely recorded before it is being normalized, 2) any extraneous loud noises or clicks have been edited from the utterance so that the largest value in the waveform corresponds to speech (usually a vowel) and, 3) the speech intensity has been relatively uniform throughout the utterance.

In situations where a human is interacting with a machine, some of these assumptions become less reasonable. From a computational perspective, it would be desirable to be able to process the signal in near or less than real-time. Another difficulty with spontaneous speech is that there are frequently spurious noises and clicks which occasionally have the largest magnitude in the signal. In this case, an utterance is not normalized with respect to speech, so the speech waveform values are subsequently weaker than normal. Finally, if the speech intensity does change somewhat over the course of the utterance, a static scaling value will result in a weaker speech signal in some portions of the utterance.

Normalization Procedure

The normalization procedure we are investigating uses a standard feedback system to compute a gain, $g[n]$. In order to provide the gain some time to adapt to changing signal conditions, there is a delay of N frames between the point where the gain is computed and where the speech signal is scaled. Currently, a frame is computed every 25 ms, and the delay is 8 frames, or 200 ms. The gain control mechanism is illustrated in Figure 1. The system input, $x[n]$, is a function of the recent short-term energy (computed with 25 ms rectangular window) of the speech signal, $s[n]$. In order to reduce the amount of change in the gain output, the input $x[n]$ is the maximum energy value within the previous N frames,

$$x[n] = \max_i s[i] \qquad n - N \le i \le n$$

The error signal, $e[n]$, is generated by comparing the scaled value $y[n]$ with a target level, $t[n]$. The scaled value is generated from the input and the previous gain value, $g[n-1]$. All values are in dB. The influence of $e[n]$ on the final gain is controlled by a scaling parameter, a, which controls the influence of the error signal on the gain.

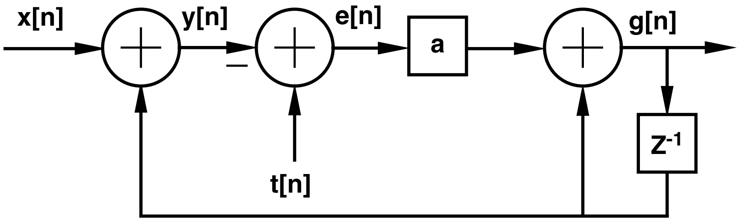

Figure 1: Normalization Schematic

Since the normalization structure assumes nothing about the signal to noise ratio, the entire signal is normalized equally whether or not the signal corresponds to speech or noise. If we could produce a value $p_s[n]$, corresponding to the probability of speech at time n, then we could modify the above algorithm. One possible modification would make the target level depend on $p_s[n]$. When $p_s[n] = 1$, then $t[n] = T_s$ the target level for speech. When $p_s[n] = 0$, then $t[n] = y[n]$ (i.e., $e[n]$= 0). In effect, when the signal is considered to be noise, the gain does not change. The net result of this argument is to make

$$t[n] = p_s[n]T_s + (1 - p_s[n])y[n]$$

or

$$e[n] = p_s[n](T_s - y[n]).$$

Thus, we make the error proportional to the probability that we have speech. Note that another alternative procedure would be to reduce the gain.

In addition to an estimate of the presence of speech in the signal, we can improve the performance of the normalization procedure if we can set a limit on the minimum allowable signal to noise ratio. This minimum can be used to set an upper limit on the gain value, and can be used for initialization purposes.

Evaluation

Since the normalization procedure we have described is a non-linear operation, it is difficult to know how to quantify its performance. We have chosen to use word recognition accuracy as one guideline. Currently, we are incorporating a version of this normalization procedure into the SUMMIT system and are evaluating the resulting performance as a function of the various parameter values.

Segmental Representations

As has been described previously [12], the SUMMIT system is based on a segmental framework. During the acoustic-phonetic analysis, a phonetic unit is mapped to a segment explicitly delineated by a begin and end time in the speech signal. Segmental frameworks have been investigated by others [3, 9] and contrast with the prevailing frame-based structure used by most HMM systems sitcom, where sequences of observation frames are assumed to be statistically independent from each other. We believe that a segmental framework offers us more flexibility than is afforded by a frame-based approach, and could ultimately lead to superior modelling of the temporal variations in the realization of underlying phonological units. It is for this reason that we base our system on such an approach.

In this section we review the current segmental framework of the SUMMIT system and present some alternatives that we have investigated. These are discussed again in the following section where we present some phonetic recognition results.

Segmental Formulation

In our framework, we try to maximize the probability of a sequence of units. For phonetic recognition, we represent the probability of a sequence of phones, $\vec{\alpha}_i = \{\alpha_{i1}, \alpha_{i2}, .., \alpha_{iN}\}$ as,

$$p(\vec{\alpha}_i) = \max_{\vec{S}_j} \prod p(\alpha_{ik}|s_{jk})p(s_{jk}) \qquad for \quad 1 \le k \le N_\alpha \quad (1)$$

where N_α is the number of phonetic units in $\vec{\alpha}_i$, $\vec{S}_j$ is the j^{th} possible sequence of N_α connecting segments traversing the utterance, $p(\alpha_{ik}|s_{jk})$ is the probability of observing a phone in a given segment, and $p(s_{jk})$ is the probability that a segment exists. During phonetic recognition, we select the sequence, $\hat{\alpha}$, that maximizes Equation 1. Thus,

$$p(\hat{\alpha}) \ge p(\vec{\alpha}_i) \qquad \forall i$$

In order to perform recognition, we need to estimate the two probability measures in Equation 1. The first term, $p(\alpha|s)$, is a set of a-posteriori classification probabilities which we will discuss later in this paper. The second term, $p(s)$, is a set of probabilities of valid segments which we will now describe in more detail.

Segment Probabilities

In order to estimate the segment probabilities, $p(s)$, we have formulated segmentation into a boundary classification problem. Let $\{b_1, b_2, .., b_k\}$ be the set of possible boundaries that might exist within segment, s_i. We define $p(s_i)$ to be the probability that all internal boundaries do not exist. To reduce the complexity of the problem, we assume that all boundaries exist independently. Thus,

$$\begin{aligned} p(s_i) &= p(\bar{b}_1, \bar{b}_2, .., \bar{b}_k) \\ &= p(\bar{b}_1)p(\bar{b}_2)...p(\bar{b}_k) \end{aligned} \qquad (2)$$

where $p(\bar{b}_j)$ stands for the probability that the j^{th} boundary does not exist. Viewing the problem as one of classification, the boundary classes can be generated by aligning a phonetic transcription with a segment space through recognition or some other procedure.

Search Space Issues

One of the disadvantages of a segmental framework is that the amount of computation associated with a search can be significant. If there are N observation boundaries in an utterance then there are 2^{N-2} possible segmentations of these N boundaries, and the number of possible segments, N_s, is $N(N-1)/2$. Conceptually, we can consider these segments as being part of a large segment network which spans from the beginning of an utterance to its end. Since phonetic classification is needed in each possible segment, the amount of computation and the amount of search throughout the segment network, can be prohibitively large. For example, if the utterance is 3 seconds long and the boundaries occur 200 times per second (our standard analysis-rate), then $N_s \approx 180,000$.

Several procedures can be adopted to improve the efficiency of a search involved in a segmental framework. First, it is clear that certain search strategies, such as those that involve dynamic programming, can reduce the amount of search. Currently, we use a modified Viterbi search for phonetic and word recognition [12]. In the following paragraphs we discuss some of the other procedures we have explored for reducing the search space.

Pruning Boundaries

If we can determine a subset of boundaries, $\mathcal{B}$, that contain all boundaries of interest, we can substantially reduce the size of the segment network. As we saw previously, the number of segments in a full network varies as the square of the number of boundaries. In our current system for instance, we use a boundary detector that, on average, locates boundaries less than one in every five frames. Thus, the number of segments is reduced by more than an order of magnitude.

Pruning Segments

There are many alternative tactics that can be used to reduce the size of the segment space. For example, Kopec and Bush used conservative duration estimates to eliminate many candidate segments [3]. In our current implementation we have used an acoustic basis for determining a set of regions, organizing the N boundaries into a hierarchical structure called a dendrogram [2]. Such a hierarchy produces $2N-3$ possible segments, which reduces the size of a segment network by a factor of approximately $\frac{N}{4}$.

Distortions

While eliminating boundaries and segments from the search space can substantially reduce the size of the search space, it can also increase the amount of distortion involved in matching a sequence of phonetic units to a segment network. In cases where there is no segment to match a phone, we say there has been a deletion of a boundary. In cases where there is no *single* segment to match a phone we say there has been an insertion of a segment. Where there is an alignment between a region and a phone there may be a certain amount of distortion involved in the alignment, which ultimately might cause a classification error.

Clearly, if the segment network is pruned there should be some mechanism for handling insertions and deletions. As the previous paragraphs have pointed out, there are various amounts of pruning that can be done, each attaining a certain level of phonetic and word recognition performance. In exploring the various possibilities our goal is to understand the behavior of the system with different amounts of pruning, and to maximize the phonetic and word recognition performance. Given these goals, we now describe some of the modifications we have made to our pruning strategies. We then describe some evaluations we made of alternative segmental approaches.

Boundary Modifications

Currently in the SUMMIT system the set of boundaries, $\mathcal{B}$, that are used are determined by a sensitive edge detector that essentially locates local maxima in a spectral derivative function [12]. This procedure appears to be quite robust since it operates on local relative changes in the speech signal. However, we have observed that there is a substantial number of boundaries located during portions of silence in the speech signal. This phenomenon has become more pronounced with our emphasis on spontaneous speech because speakers tend to false start and hesitate more often than when they are reading. As a result, there can be a significant silence period and/or non-linguistic sounds at the beginning and the end of the sentence. One consequence of this is that the system spends a large amount of time analyzing the many segments produced during periods of silence.

Aiming at reducing computational load in different parts of the system, we have been investigating techniques to prune the silence regions. We have incorporated into SUMMIT a simple algorithm which uses scores for speech and silence based on the distributions of eight principle components of the mean rate response outputs of an auditory model [11]. We trained the system by collecting histograms of parameter distributions for phonetically transcribed utterances from a spontaneous-speech database[13]. The probability of speech is computed on a frame by frame basis, after some temporal smoothing. A two-state Markov model contains a-priori transition probabilities that are incorporated into the score in order to delineate long regions of silence or speech. We are in the process of evaluating the effect of this procedure on word recognition accuracy.

Dendrogram Modifications

In the hierarchical clustering procedure currently used in the SUMMIT system we require a metric, D, to compute a distance between two adjacent regions. Specifically, let $\vec{a}_i$ be the acoustic vector corresponding to the i^{th} region. Then $D_i = D(\vec{a}_i, \vec{a}_{i+1})$ corresponds to the distance between the i^{th} and $i+1^{st}$ regions. In our clustering procedure we merge two adjacent regions when,

$$D_i < min\{D_{i-1}, D_{i+1}\}$$

The current procedure uses a weighted Euclidean distance metric. The acoustic vector is an average spectral vector of the segment. We have observed two problems that occur in the resulting dendrograms. Due to the Euclidean metric, little weight is given to correlation across adjacent channels in the spectral representation. Thus, the acoustic structure cannot always distinguish similar sounds from those whose spectral shape is significantly different. Second, local extrema in the representation were not adequately reflected in the resulting dendrogram structure. The combined effects of these phenomena was to produce a large phonetic alignment distortion for certain sequences of sounds.

We attempted to address these issues by first translating the spectral representation using principle component analysis. Each dimension was then normalized by its mean and variance. An additional rotation was made based on the average within-class variance of each dimension. These variances were generated from aligned phonetic transcriptions and were intended to equalize the contribution of each component dimension to the overall distance metric. We explored several distance metrics using this representation and found that an effective one was based on a normalized dot-product,

$$\hat{D}(\vec{a}_i\vec{a}_{i+1}) = 1 - \frac{\vec{a}_i \cdot \vec{a}_{i+1}}{|\vec{a}_i||\vec{a}_{i+1}|}$$

which is proportional to a Euclidean distance between the normalized acoustic vectors.

The problem with the local extrema was addressed by incorporating more information into the distance metric. Specifically, we computed difference vectors, $\vec{d}_i = \vec{a}_i - \vec{a}_{i+1}$ and computed $\hat{D}(\vec{d}_i, \vec{d}_{i+1})$. The final distance metric, D^*, used a combination of both metrics,

$$D_i^* = \hat{D}(\vec{d}_{i-1}\vec{d}_i)\hat{D}(\vec{d}_i\vec{d}_{i+1})\hat{D}(\vec{a}_i\vec{a}_{i+1})$$

Evaluation

The evaluation of various segmental representations can take many forms. The first analysis we performed aligned a phonetic transcription with the segment network. The alignment procedure mapped a phonetic token to the closest region that overlapped by at least 50%. If no region was found a deletion or insertion was made. Table 1 summarizes the statistics of the various representations we have described on 150 TIMIT utterances. These utterances contained 5636 phonetic tokens. From the Table we see that the modified hierarchical procedures reduce the insertion rate of the current representation by more than one third while the deletion rates are slightly reduced. Finally, a full segment network created from the entire boundary set $\mathcal{B}$ has essentially no insertions. The minimum deletion rate is just under 2%.

Segment	Deletion (%)	Insertion (%)
SUMMIT	3.5	5.7
$\hat{D}$	3.5	4.7
D^*	3.0	3.7
$\mathcal{B}$	1.9	0

Table 1: Phonetic alignment performance.

Acoustic-Phonetic Analysis

In the previous section we outlined the segmental representation that is being used in the SUMMIT system and described some alternatives to the current implementation. In this section we describe some experiments we have performed to explore alternative methods for phonetic classification based on multi-layer perceptrons (MLP). These experiments involve both phonetic classification and recognition. They compare classification techniques as well as segmental representations. In the next sections we describe the task, and speech corpus used for the experiments, as well as the MLP classifier and data representations that were used. This is followed by a description of the phonetic classification and finally the phonetic recognition experiments.

Task and Corpus

All experiments were based on the sx sentences from 350 speakers of the TIMIT database [4]. 1500 sentences from 300 speakers were used for training, and 250 sentences from the remaining 50 speakers were used for testing. As summarized in Table 2, there were a total of 55,000 phonetic tokens in the training data and 9,000 tokens in the testing data. There were 38 phonetic labels used which represented 14 vowels, 3 semivowels, 3 nasals, 8 fricatives, 2 affricates, 6 stops, 1 flap, and silence. This particular set was chosen because it has been used in other evaluations both within and outside our research group [6, 12].

Training	Test	Training	Test
Speakers (M/F)	Speakers (M/F)	Tokens	Tokens
300 (216/84)	50 (33/17)	55,000	9,000

Table 2: Corpus used for the experiments.

MLP Classifier

Recently, we have been investigating the use of multi-layer perceptrons for phonetic classification. Our work was motivated by the belief that these networks might offer a flexible framework for us to utilize our improved, albeit incomplete, speech knowledge. Until recently, our study was performed on the constrained task of classifying the 16 vowels in American English, spoken by many speakers and excised from continuous speech [8]. Our encouraging results suggested that MLP is a promising technique worthy of further investigation.

We extended our work to one of classifying 38 vowels and consonants. In moving to this larger phonetic classification problem, we discovered some major problems in training the network. In this section, we will describe these problems and suggest some procedures to overcome them.

Initialization

Without any a priori knowledge, the connection weights of MLP are often randomly initialized. Since the transition region of the sigmoid function is relatively narrow while the saturation regions are relatively wide, randomly initializing the network can have the adverse effect of causing most of the basic units to operate in the saturation regions of the sigmoid function, where learning is slower than in the transition region.

Let $z_i = \sum_j w_{ij}x_j$, where z_i is the input to unit i. If we assume that the weights, w_{ij}, are randomly initialized such that they are uncorrelated with zero mean and constant variance. It can be shown that [8]:

$$E[z_i] = \sum_j E[w_{ij}]E[x_j] = 0 \quad \text{and} \quad \sigma_{z_i}^2 = \sigma_w^2 \sum_j E[x_j^2]. \tag{3}$$

Thus although the expected value of the input to the sigmoid of a basic unit, $E[z_i]$, is zero, the variance, $\sigma_{z_i}^2$, depends on the variance of the random weights, σ_w^2, as well as the magnitudes and the number of dimensions of the input vectors. If $\sigma_{z_i}^2$ is large, many basic units may operate in the saturation regions.

Normalization of Inputs

Several procedures have been suggested to enable the basic units to operate initially in the transition regions. By initializing the network with small random weights or biasing the inputs, $\sigma_{z_i}^2$ can be reduced [1]. A method called center initialization has also been suggested that guarantees all the basic units initially operate at the center of the transition region, where learning is fastest [8].

However, as training proceeds, the connection weights are changed, resulting in $E[z_i]$ and $\sigma^2_{z_i}$ depend progressively more on the set of input vectors, $\vec{x}$. If $E[x_j]$ and $\sum_j E[x_j^2]$ are large, the hidden units may get driven well into the saturation regions, hindering the learning capability of the network.

Let $\{\vec{\mathbf{s}}\}$ denote the set of training samples, where $\vec{\mathbf{s}} = [s_1, s_2, \ldots]^t$ is the speech vector for each phoneme token. Furthermore, let

$$x_j = \frac{(s_j - \bar{s}_j)}{\gamma \sigma_j}, \tag{4}$$

where σ_j is the standard deviation of s_j, $\bar{s}_j$ is the mean of s_j over all the training tokens, and γ is a positive constant. Thus, $E[x_j] = E[z_i] = 0$. Assuming $\gamma\sigma_j > 1$,

$$\sigma^2_{z_i} = \sigma^2_w \sum_j E[\frac{(s_j - \bar{s}_j)}{\gamma\sigma_j}]^2 < \sigma^2_w \sum_j E[s_j]^2. \tag{5}$$

Thus by subtracting and normalizing the input patterns according to Equation (4), the hidden units may operate more often in the transition region of the sigmoid function.

Adaptive Gain

Although Equation (4) provides a mechanism to increase the learning capability of the network, it requires that $\bar{s_j}$ and σ_{s_j} be computed before the network is trained. In this section, we discuss a different technique to deal with the learning capability of the network.

During training, the connection weights are usually modified according to $\Delta w_{ij} = \eta \delta_i x_j$, where η is the gain term, and δ_i is the error signal for unit i. (For simplicity, the momentum term is ignored in the following analysis.) Thus $|\Delta w_{ij}|$ depends on $|x_j|$, which means the training procedure pays more attention to inputs with larger magnitudes than to inputs with smaller magnitudes.

Alternatively, η can be chosen to be adaptive. Specifically,

$$\eta = \frac{\eta_0}{\sum_j |x_j|}, \tag{6}$$

where η_0 is a small positive constant. Thus

$$|\Delta w_{ij}| = \frac{\eta_0}{\sum_j |x_j|} |\delta_i|\ |x_j|, \quad \text{and} \quad \sum_j |\Delta w_{ij}| = \eta_0 |\delta_i|. \tag{7}$$

As a result, the total change in the connection weights to a hidden unit is independent of the input, $\vec{\mathbf{x}}$, thus allowing the training procedure to pay similar attention to all input vectors.

Boundary Classification

In our segmental framework formulated in Equation 1, the main difference between classification and recognition is the incorporation of a probability for each segment, $p(s)$. As described previously in Equation 2, we have simplified the problem of estimating $p(s)$ to one of determining the probability that a boundary exists, $p(b)$. Currently in the SUMMIT system, boundary classification is based on the height attained by a boundary in the dendrogram. A small VQ codebook of size 12 is used to quantize the spectral average of each segment, and distributions are collected and parameterized for each possible context.

Since one of the segment networks we considered was not based on a dendrogram, an alternative classification procedure for the boundaries was adopted. In this procedure, a MLP with two output units was used, one for the valid boundaries and the other for the extraneous boundaries. By referencing the time-aligned phonetic transcription, the desired outputs of the network can be determined. In our current implementation, the probability of a detected boundary, $p(b_i)$, is determined using four abutting segments. Let t_i stand for the time at which b_i is located, and s_i stand for the segment between t_i and t_{i+1}, where $t_{i+1} > t_i$. The boundary probability, $p(b_i)$, is then determined by using the acoustic measurements in s_{i-2}, s_{i-1}, s_i, and s_{i+1} as inputs to the MLP.

Data Representation

There were two representations used as input for the MLP classifier. The first representation was identical to the SUMMIT system, and consisted of 82 acoustic attributes. The attributes were generated automatically by a search procedure that uses the training data to determine the settings of free parameters of a set of generic property detectors using an optimization procedure[10]. The second representation consisted of a vector of three average spectra which corresponded to the left, middle, and right thirds of a segment. The spectra were the mean-rate and synchrony outputs of a 40 channel auditory model [11]. Thus, there were 120 points used for each representation. Finally, segment duration was also included.

Experimental Results

Phonetic Classification

The first experiments which were performed were based on phonetic classification. In these tests the system classified a token taken from a phonetic transcription that had been aligned with the speech waveform. Since there was no detection involved in these experiments only substitution errors were possible. As has been reported previously, the baseline speaker-independent classification performance of SUMMIT on the testing data was 70% [12]. The performance of the MLP classifier using the same input representation was 74%. In the second set of classification experiments the representation was based on the spectral outputs described previously. Four experiments were performed using 1) the synchrony outputs, 2) the mean-rate outputs, 3) the synchrony and mean-rate outputs and 4) the synchrony and mean-rate outputs and segment duration. The results of all experiments have been summarized in Table 3. None of these alternative representations was able to achieve the same level of performance as the SUMMIT acoustic attributes.

Boundary Classification

We have evaluated the MLP boundary classifier using the training and testing data described earlier. The inputs to

Classifier	Representation	Performance (%)
Baseline	attributes	70
MLP	attributes	74
MLP	SYN	65
MLP	MR	68
MLP	SYN+MR	70
MLP	SYN+MR+DUR	72

Table 3: Phonetic classification comparing the baseline and MLP classifiers, and acoustic representations. The representations are the 82 acoustic attributes used in SUMMIT, the synchrony envelopes (SYN), mean-rate response (MR), and duration (DUR).

the network are the averages of the mean rate response in the four abutting segments, resulting in a total of 160 input units. By using 32 hidden units, the network can classify 87% of the boundaries in the test set correctly.

Phonetic Recognition

The results of the phonetic recognition experiments are shown in Table 4. The baseline performance of the system is 47%, including substitution, deletion, and insertion errors. All of the MLP based recognizers showed an improved performance over the baseline system. The modified dendrogram showed a 3% improvement over the baseline dendrogram. The segment networks based on the entire boundary set, $\mathcal{B}$, which were naturally larger than the dendrogram networks, showed the largest overall improvement. In order to reduce the amount of computation, these networks were pruned based on conservative duration constraints, so they contained only twice as many regions, on average, as the dendrogram networks.

Classifier	Segment	Performance (%)
Baseline	Baseline	47
MLP	Baseline	50
MLP	D^*	53
MLP	$\mathcal{B}$	55

Table 4: Phonetic recognition.

Summary

In this paper we have described a normalization procedure that we are investigating that requires a small amount of look-ahead in the speech signal. We have also reviewed the segmental representations that are used in SUMMIT, and documented our current investigations to balance a moderate computational load with increased performance. Finally, we have reported our phonetic classification and recognition experiments with multi-layer perceptrons. These experiments showed that we could improve our phonetic classification rate by 4% through the use of an MLP classifier and a set of automatically determined acoustic attributes. In addition, we were able to improve our phonetic recognition rate by 8%, by combining the MLP classifier with an alternative segment network that uses twice as many segments as that of our baseline network. In the future, we plan to continue this line of investigation at the word recognition level.

Acknowledgements

Upendra Chaudhari provided programming help in the development of the normalization procedure we described in this paper.

References

[1] Burr, D.J., “Experiments on Neural Net Recognition of Spoken and Written Text,” *IEEE Trans. Acoust. Speech and Sig. Proc.,* Vol. 36, July, 1988.

[2] Glass, J.R., and V.W. Zue, ”Multi-level Acoustic Segmentation of Continuous Speech, *Proc. ICASSP-88*, pp. 429–432, New York, 1988.

[3] Kopec, G.E., and M.A. Bush, “Network-based Isolated Digit Recognition Using Vector Quantization,” *IEEE Trans. Acoust. Speech and Sig. Proc.*, Vol. ASSP-23, pp. 850–867, 1985.

[4] Lamel, L.F., Kassel, R.H., and Seneff, S., “Speech Database Development: Design and Analysis of the Acoustic Phonetic Corpus,” *Proc. DARPA Speech Recognition Workshop,* pp. 100–109, 1986.

[5] Lee, K.-F., *Automatic Speech Recognition: The Development of the SPHINX System*, Kluwer Academic Publishers, Boston, 1989.

[6] Lee, K.-F., and Hon, W.-H., “Speaker-independent Phone Recognition Using Hidden Markov Models,” *IEEE Trans. Acoust. Speech and Sig. Proc.,* Vol. 37, No. 11, pp. 1641–1648, 1989.

[7] Leung, H.C., and V.W. Zue, “Some Phonetic Recognition Experiments Using Artificial Neural Nets,” *Proc. ICASSP-88*, pp. 422–425, New York, 1988.

[8] Leung, H.C., *The Use of Artificial Neural Networks of Phonetic Recognition,* Ph.D. Thesis, Massachusetts Institute of Technology, 1989.

[9] Ostendorf, M., and S. Roukos, “A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition,” *IEEE Trans. Acoust. Speech and Sig. Proc.*, Vol. 37, No. 12, pp 1857–1869, 1989.

[10] Phillips, M.S., ”Automatic Discovery of Acoustic Measurements for Acoustic Classification,” *J. Acoust. Soc. Amer.*, Vol. 84, S216, 1988.

[11] Seneff, S. “A Joint Synchrony/Mean-rate Model of Auditory Speech Processing,” *Proc. J. of Phonetics*, Vol. 16, pp. 55–76, 1988.

[12] Zue, V., J. Glass, M. Phillips, and S. Seneff, “The MIT SUMMIT Speech Recognition System: A Progress Report,” *Proceedings of DARPA Speech and Natural Language Workshop,* February, 1989.

[13] Zue, V., N. Daly, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, S. Seneff, and M. Soclof, “The Collection and Preliminary Analysis of a Spontaneous Speech Database,” *Proc. Second DARPA Speech and Language Workshop,* Morgan Kaufman and Co., 1989.