UNIFIED ACOUSTIC MODELING FOR CONTINUOUS SPEECH RECOGNITION

Rathinavelu Chengalvarayan

Speech Processing Group,
Lucent Speech Solutions Department
Lucent Technologies, Naperville, IL 60566, USA
Email: rathi@lucent.com

ABSTRACT

Usually the speech and the silence models are trained together depending upon the type of recognition task. For example, if the recognition task is only on connected-digits then the corresponding digit models are built using only the connected-digit training corpus. Similarly for large-vocabulary recognition tasks, the subword or the phoneme models are generated using only the subword training set. Further the alphabet models are separately trained using the alphabet training data for letter recognition. In certain applications the developer needs to perform mixed-mode operations like alphabet followed by digits, digits succeeded by keywords, letters preceded by keywords etc. So there is a need to robustly design a speech recognizer for such kind of specific applications. In that context, we propose several acoustic modeling techniques to improve the unified model performance for applications that require mixed-mode operations.

1. INTRODUCTION

Generally we train the speech and silence models together depending upon the type of recognition task. Since the spoken keywords may contain silences in between them or background noises at the begining or at the end of the utterance. So it is very difficult to train the speech model alone without using the silence model in the core training process. For example, if the recognition task is only on connected-digits then we build the corresponding digit models using either connected-digit or isolated-digit training set [1]. Similarly for large-vocabulary tasks, either context-dependent triphone models or context-independent subwords are built using only the subword training set [2, 3, 12]. Further, the alphabet models are separately trained using alphabet training data for letter recognition [9, 4, 6]. Altogether we have three different silence models that are trained for three different recognition tasks.

In certain applications such as VoiceXML and voice name dialing the automatic speech recognizer (ASR) needs to perform mixed-mode operations like alphabet followed by digits ("N2L3G1" which is the zip code for University of Waterloo in Canada), keywords followed by digits ("call area code 630 and 717-9597"), letters followed by keywords ("IBM or ABC corporation") etc. Some of the common applications of mixed-mode operations will be systems that allow customers to access information in a company database over the phone. Some simple examples include movie, weather, and traffic information phones, order tracking applications, e-mail readers and personal information managers. More complex applications include speech recognition enabled call centers for catalog shopping, airline reservations, stock trades and financial services management [7, 15, 10].

Moreover there is a need to evaluate the system for such mixed-mode ASR applications. In that context, we propose several new acoustic modeling techniques to improve the unified model performance for mixed-mode operations. We also show that by mixing the silence models among different recognition tasks drastically reduces the ASR performance, for instance if we use connected digit silence model for subword recognition task, the performance on city-name and company-name recognition drops and vice-versa. Preliminary experimental results show that by having a unified common silence model for all three independent recognizers, one can achieve a better string accuracy than using a specific silence model that are trained over a particular task-dependent environment.

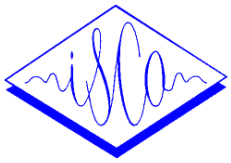
2. DISCRIMINATIVE TRAINING

We have tried two methods for obtaining estimates of the HMM parameters, namely the conventional maximum likelihood estimation (MLE) and a more effective minimum string error (MSE) algorithms. For MLE training, the segmental k-means training procedure was used [11]. An example of discriminative training is the MSE training algorithm which directly applies discriminative analysis techniques to string-level acoustic model matching, thereby implementing minimum error rate training at the string level [14]. In MSE training, the string level acoustic modeling is based on the string model of the correct word string and the string models of the N most confusable word strings obtained by using a fast tree-trellis based N-best search [13]. Let $S = W_1, ..., W_{l_s}$ be an arbitrary word string. Given the model set Λ, the optimal state sequence Θ_S is a function of the observation O and the word string S. The top N best string hypotheses $S_1, ..., S_N$ can be defined inductively as follows:

$$S_1 = \arg\max_{S} \log f(O, \Theta_S, S|\Lambda), \quad (1)$$

$$S_k = \arg\max_{S \neq S_1,...,S_{k-1}} \log f(O, \Theta_S, S|\Lambda) \quad (2)$$

In MSE training, these string level acoustic training tokens are incorporated into a set of discriminant functions,



6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

specially designed for representing string errors. This is achieved through the following four steps:

1. Discriminant function in MSE training is defined as

$$g(O, S_k, \Lambda) \quad = \quad \log\ f(O, \Theta_{S_k}, S_k|\Lambda), \qquad (3)$$

where $O = O_1, ..., O_T$ is the observation feature vector sequence of length T from the training speech samples, S_k is the k-th best string, Λ is the HMM set used in the N-best decoding, Θ_k is the optimal state sequence of the k-th string given the model set Λ, and $\log\ f(O, \Theta_{S_k}, S_k|\Lambda)$ is the related log-likelihood score on the optimal path of the k-th string. For the correct string S_c, the discriminant function is given by

$$g(O, S_c, \Lambda) \quad = \quad \log\ f(O, \Theta_{S_c}, S_c|\Lambda), \qquad (4)$$

where S_c is the correct string, Θ_c is the optimal alignment path and $\log\ f(O, \Theta_{S_c}, S_c|\Lambda)$ is the corresponding log-likelihood score. These discriminant functions depend on both the model set Λ and the particular word string S under consideration. In ML-based training approach, the model parameter estimate is based only on the correct string model given in equation (4). The discriminative information existing in the competing string models described in equation (3) is generally not used.

2. The misclassification measure is determined by

$$d(O, \Lambda) = -g(O, S_c, \Lambda) + \log\left(\frac{1}{N-1}\sum_{S_k \neq S_c} e^{g(O, S_k, \Lambda)}\right)$$

which provides an acoustic confusability measure between the correct and competing string models.

3. The loss function is defined as

$$l(O, \Lambda) \quad = \quad \frac{1}{1 + e^{-\gamma d(O,\Lambda)}}, \qquad (5)$$

where γ is a positive constant, which controls the slope of the sigmoid function.

4. The model parameters are updated sequentially according to the generalized probabilistic descent algorithm such that

$$\Lambda_{n+1} \quad = \quad \Lambda_n - \epsilon \nabla l(O, \Lambda), \qquad (6)$$

Λ_n is the parameter set at the nth iteration, $\nabla l(O, \Lambda)$ is the gradient of the loss function for the training sample O which belongs to the correct class c, and ϵ is a small positive learning constant.

In this paper, we report only the results obtained by sequential training. During the model training phase, we call one complete pass through the training data set as an epoch. For the case of string-by-string training, model parameters are updated several times over an epoch.

3. SPEECH DATA

The experimental results are based on a continuous speech database containing speech utterences recorded over the telephone network in a U.S. wide data collection covering the different dialect regions. Male and female speakers were fairly equally represented. The connected-digit training and testing set are valid digit strings, totaling 7282 and 13114 strings for training and testing, respectively. The subword training set consists of 9865 generic phrases and the city-name testing set contains 3620 spontaneous utterences of *city name* followed by either a state or a country name, for example, *Beijing China*. There are about 448 unique city-names in this testing set. The company-name testing set contains 11552 spontaneous utterences of *company names* from 843 speakers and 6923 unique names. The alphabet corpus contains about 12638 strings for training and 3629 strings for testing.

4. FEATURE EXTRACTION

The speech input is sampled at 8kHz and preemphasized using a first-order filter with a coefficient of 0.95. The samples are blocked into overlapping frames of 30 msec in duration, where the overlap is set to 20 msec. Each frame is windowed with a Hamming window and then processed using a 10th-order LPC analyzer. The LPC coefficients are then converted to cepstral coefficients, where only the first 12 coefficients are retained. The basic recognizer feature set consists of 36 features that includes the 12 liftered cepstral coefficients and their first and second order derivatives [1]. Besides the cepstral based features, the normalized energy contour and its first and second order time derivatives are also computed. Thus, each speech frame becomes represented by a vector of 39 features. Note that the computation of all the higher order coefficients is performed over a segment of five frames. Since the signal has been recorded under various telephone conditions and with different transducer equipment, each cepstral vector was further processed using the one-level cepstral mean subtraction method in order to reduce the effect of channel distortion [5].

5. HMM ARCHITECTURES

The connected-digit recognizer models each word in the vocabulary by a set of left-to-right continuous mixture density HMM using context-dependent head-body-tail models [1]. Each word in the vocabulary is divided into a head, a body, and a tail segment. To model inter-word coarticulation, each word consists of one body with multiple heads and multiple tails depending on the preceding and following contexts. In this paper, we model all possible inter-word coarticulation, resulting in a total of 276 context-dependent sub-word models. Both the head and tail models are represented with 3 states, while the body models are represented with 4 states, each having 8 mixture components. This configuration results in a total of 276 models, 837 states and 3904 mixture components.

The subword model set used in the recognition consists of 41 context independent units [8]. Each subword is modeled by a three state left-to-right continuous density HMM with only *self* and *forward* transitions. A mixture of Gaussians

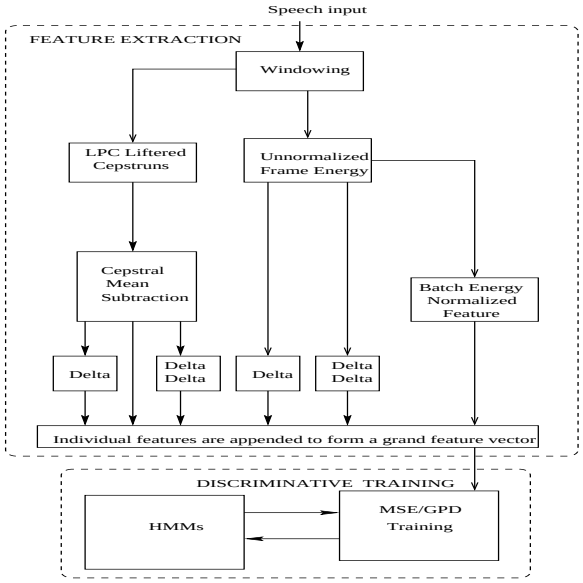


Figure 1. Block diagram of feature analysis using signal conditioned MSE training.

with diagonal covariances is employed to estimate the density function for each state. A maximum of 16 mixtures per state is allowed [2, 8]. The silence/background is modeled with a single state, 32 Gaussian mixture HMM. Furthermore no transition probabilities are used. The lexical representations of the sentences are obtained by preprocessing the sentence orthographic transcriptions through a text-to-speech front end. The grammar used in the recognition is the standard namelist grammar (pick a best candidate from a list of N recognized candidates).

The acoustic models are continuous density, context independent, left-to-right HMMs with each letter of the alphabet being modeled as a separate unit. Vowels and consonants are all modeled with 10 states, where each state is modeled as a mixture of 8 Gaussians and the speech background is modeled with a single state, 32 mixture silence HMM [4, 9]. All the initial model sets were trained using one-iteration of conventional maximum likelihood training procedure [11]. We then applied five iterations of MSE training to the initial boot model with null grammar and the number of competing string models was set to four. The MSE training as discussed in section 2 is applied to each training utterance with or without silence skipping depending upon the model building process. In some special cases the initial speech models are updated during MSE training but the silence model is untouched. Silence is modeled with a single state model having 32 mixture components. Training included updating all the parameters of the model, namely, means, variances, mixture gains and duration statistics. The integration of feature analysis and the modeling process with overlapping in the signal conditioning is illustrated in Figure 1.

6. UNIFIED ACOUSTIC MODELING

Different ways of building an unified acoustic models for mixed-mode application are described below:

Model	Conn_Dig	City_Name	Comp_Name
BSM	93.80%	96.22%	92.65%
CSM	93.80%	94.53%	88.31%
SSM	88.12%	96.22%	92.65%
ASM	90.01%	95.53%	90.91%
USM	90.88%	96.34%	92.50%
GSM	92.56%	95.43%	91.10%
SSM	93.53%	96.49%	92.63%

Table 1. string accuracy for an unknown-length grammar-based U.S. English connected-digit and large-vocabulary recognition tasks as a function of model type.

- Benchmark Silence Model (**BSM**): Each model set has its own silence model built using five iterations of MSE algorithm. Totally we have three silence models, one for connected digits, one for subword tasks and one for alphabet recognition. For each task one can explicitly pick the appropriate silence model so that the best result can be obtained.
- Connected-Digit Silence Model (**CSM**): We built the individual model set with a common connected digit silence. This is same as BSM but with a single connected digit silence model for all three recognition tasks.
- Subword Silence Model (**SSM**): We built the individual model set with a common subword silence. This is same as CSM but with a global subword silence.
- Alphabet Silence Model (**ASM**): This is same as CSM but with a global alphabet silence model.
- Grammar-Based Silence Model (**GSM**): This is same as BSM but the silence model is picked-up based on the grammar constraint. By default it uses the connected-digit silence for decoding. If there is a transition from digits to subword then the subword silence model is picked during likelihood calculation. Similarly if there is a transition from digits to alphabet then the alphabet silence is picked up during decoding. So GSM is different from BSM in the sense that GSM doesn't need an explicit silence selection, it does automatically through the incoming grammar constraint. Sometimes we call this selective method as context-dependent silence picking.
- Silence Skip Model (**SSM**): The adaptation algorithm consists of the following two steps:
  1. *Step 1*: Train an universal silence model from an uniform pool of common segments obtained by using all the available connected-digit, subword, and alphabet training data. The background (silence) is modeled with a single state, 32 Gaussian mixture HMM using a few iterations of K-means clustering algorithm [11].
  2. *Step 2*: Perform five iterations of MSE training algorithm using the universal silence model that was built in step 1 on each set of speech models independently. In this study we have three sets of speech models: 276 head-body-tail connected-digit models; 41 subword models; and 27 alphabet

Model	Alpha_Let	Overall	Err
BSM	51.01%	89.43%	0%
CSM	49.35%	87.49%	-15%
SSM	46.24%	86.79%	-20%
ASM	51.01%	87.29%	-17%
USM	49.02%	88.09%	-11%
GSM	50.15%	88.21%	-10%
SSM	52.64%	90.10%	+7%

Table 2. string accuracy for an unknown-length grammar-based U.S. English letter recognition task as a function of model type and the corresponding string error rate reduction (Err) in % when compared to BSM.

models; Make a note that the silence model is not updated during MSE training process. Finally all the individual models are tested using the MSE skipped MLE built universal silence model on various task-dependent environments.

- Universal Silence Model (**USM**): It is same as SSM, but the common or universal silence model is also updated along with other speech models during MSE training as exemplified in section 2.

7. EXPERIMENTAL RESULTS

The Tables 1 and 2 show the string accuracy of different unified acoustic models with various silence training/picking algorithm. We see that by using a task-dependent silences, the ASR performance drops due to mismatched conditions (subword silence is used in connected digit recognition task and vice-versa). The universal USB and GSM are better than the CSM, SSM and ASM models but it is inferior to the BSM models. Make a note that the benchmark BSM is the second best model since we force the best silence during decoding. The SSM model based on silence skipping during MSE outperforms all the other models and we observed a string error rate reduction of 7% when compared to the second best BSM model. The main benefit of using SSM technique is that we can train the individual model sets in many different parallel machines as opposed to sequential processing in USM and hence the ultimate model building execution time becomes much faster.

8. CONCLUSIONS

We proposed several acoustic modeling techniques to improve the unified model performance for applications that require mixed-mode operations. We observed that by using a task dependent silences, the ASR performance drops due to mismatched training and testing conditions (when subword silence is used in connected digit recognition task, the overall string error rate drops by about 20%). The test results showed that by having a unified common silence model, which is MLE-trained from all the available training segments and thereafter it is untouched during MSE training, one can achieve a better ASR performce than by using a task-specific silence.

REFERENCES

[1] R. Chengalvarayan, "A comparative study of hybrid modelling techniques for improved telephone speech recognition", *Proc. ICSLP*, pp. 313-316, 1998.

[2] R. Chengalvarayan, "On the use of normalized LPC error towards better large vocabulary speech recognition systems," *Proc. ICASSP*, pp. 17-20, 1998.

[3] 3) R. Chengalvarayan and L. Deng, "Use of generalized dynamic feature parameters for speech recognition", *IEEE Trans. on Speech and Audio Processing*, Vol. 5, No. 3, pp. 232-242, 1997.

[4] R. Chengalvarayan, "Discriminative hidden Markov models using vector-valued dynamic weighting parameters for alphabet recognition", *The 4th World Multi-Conference on Systemics, Cybernetics and Informatics*, Vol. 5, pp. 37-40, 2000.

[5] R. Chengalvarayan, "On-line cepstral normalization for cellular hands-free speech recognition", *IEEE International Symposium on Intelligent Signal Processing and Communication Systems*, 2000.

[6] P.C. Loizou and A.S. Spanias, "High-performance alphabet recognition", *IEEE Trans. on Speech and Audio Processing*, Vol. 4, No. 6, pp-430-445, 1996.

[7] S. Ihnen, "VoiceXML: A developer's view", *Speech Technology Magazine*, Vol. 4, pp. 8-12, 2000.

[8] R.A. Sukkar and C-H. Lee, "Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition", *IEEE Trans. on Speech and Audio Processing*, Vol. 4, No. 6, pp. 420-429, 1996.

[9] C. Mitchell and A. Setlur, "Improved spelling recognition using a tree-based fast lexical match", Proc. ICASSP, pp. 597-600, 1999.

[10] M. Gandhi and J. Jacob, "Natural number recognition using MCE trained inter-word context dependent acoustic models", Proc. ICASSP, pp. 457-460, 1998.

[11] B-H. Juang and L. Rabiner, "The segmental K-means algorithm for estimating parameters of hidden Markov models", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 38, No. 9, pp. 1639-1641, 1990.

[12] C-H. Lee, B-H. Juang, W. Chou and J.J. Molina-Perez, "A study on task-independent subword selection and modeling for speech recognition", Proc. ICSLP, pp.1820-1823, 1996.

[13] F.K. Soong and E.F. Huang, "A tree-trellis based fast search for finding the N-best sentence hypotheses in continuous speech recognition", *Proc. ICASSP*, pp. 705-708, 1991.

[14] S. Katagiri, B-H. Juang and C-H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2345-2373, 1998.

[15] Y. Muthusamy, R. Agarwal, Y. Gong and V. Viswanathan, "Speech-enabled information retrieval in the automobile environment", Proc. ICASSP, pp. 2259-2262, 1999.