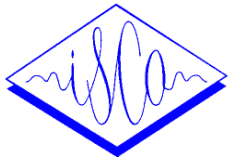

7th International Conference on Spoken Language Processing [ICSLP2002]
Denver, Colorado, USA
September 16-20, 2002

IMPROVEMENTS TO THE IBM AURORA 2 MULTI-CONDITION SYSTEM

George Saon and Juan M. Huerta

IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598
E-mail: {saon,huerta}@watson.ibm.com, Phone: (914)-945-2985

ABSTRACT

In this paper we describe some recent improvements to the performance of the Aurora 2 noisy digits speech recognition system for the matched training and test condition. The algorithms that we used pertain to discriminant acoustic modeling based on the Maximum Mutual Information (MMI) criterion, non-linear speaker/channel adaptation through probability distribution function matching. In addition, we revisited our last year's baseline system and improved its performance through cross-word context dependent modeling and Gaussian mixture components selection using the Bayesian Information Criterion (BIC). The aggregated result is 93.3% word accuracy for the multi-condition training data scenario.

1. INTRODUCTION

We report on some improvements to the multi-condition speech recognition system for the Aurora 2 noisy digits database. In section 2 we present an overview of last year's system characteristics, in section 3 we show some basic improvements to the baseline system such as cross-word context dependent modeling and Gaussian selection using the Bayesian Information Criterion and in section 4 we describe two new techniques for this year's system: MMI acoustic model training and non-linear adaptation through probability distribution function matching. Finally, in section 5 we present a summary of the results obtained.

2. LAST YEAR'S SYSTEM

For a complete description of the system we refer the reader to [5]. As a reminder, we briefly recall the main components in terms of front-end processing, acoustic models and adaptation/compensation techniques mainly in order to highlight the differences with our current setup.

The multi-condition system uses 13 Mel frequency-warped cepstral coefficients (MFCC) computed every 10ms from a 24-filter Mel filterbank. Every 9 consecutive cepstral frames are spliced together and projected down to 39 dimensions using linear discriminant analysis (LDA). The range of this transformation is further diagonalized by means of a maximum likelihood linear transform (MLLT). The acoustic models comprise 22 context-independent phones, each phone being modeled by a 3-state HMM. The output distributions for the 66 sub-phonetic classes have 3.2K diagonal covariance Gaussian mixture components. The Viterbi decoding of the utterances is performed without any form of pruning with the insertions/deletions ratio controlled by a word-insertion penalty.

Acoustic adaptation is performed entirely on the features (the models are not altered in any way) by means of a maximum likelihood feature space regression (called FMLLR). We have extended this transform to the case of a projection on the features (called FMLLR-P) as opposed to a square transformation and shown its superiority to the square case for this task.

3. BASELINE SYSTEM

3.1. Acoustic models

The current recognition system uses a phonetic representation of the words in the vocabulary (with an alphabet of 22 phones). Each phone is modeled with a 3-state left-to-right HMM. Further, we identify the variants of each state that are acoustically dissimilar by asking questions about the phonetic context (within an 11-phone window) in which the state occurs. The questions are arranged hierarchically in the form of a decision tree, and its leaves correspond to the basic acoustic units that we model. The system uses 127 leaves and there are on average 3.5 cross-word context-dependent variants for each word in the lexicon. The cross-word context dependency occurs only on the left of a word, i.e. the acoustic units are independent with respect to the following words.

The output distributions for the 127 leaves are given by a mixture of at most 128 diagonal covariance Gaussian components totaling around 9.5K Gaussians. The exact number of Gaussians for each leaf was determined using the Bayesian Information Criterion (or BIC) [2]. The penalty factor for the log-likelihood of the data for BIC was set to 1. Compared to last year's system, we almost tripled the number of Gaussian densities and this turned out to be beneficial as was also pointed out in [3].

3.2. Search strategy

We perform unpruned Viterbi decoding on a static (precompiled) HMM network obtained by expanding the words in the acoustic vocabulary in terms of their phones (and leaves). The network contains 164 emitting states and 65 null states used to enforce the contextual constraints (which context-dependent variants of words can follow each other in the graph) [8]. Another difference with last year's system is that we set the word insertion penalty to -15 or, which is equivalent in terms of the Viterbi computation, we down-scaled the acoustic likelihoods by 1/15 and set the insertion penalty to -1.

4. NEW TECHNIQUES FOR AURORA

4.1. MMI training

MMI training was originally proposed in [1] as an alternative to ML and maximizes the mutual information between the training word sequences and the observation sequences. The MMI criterion increases the a posteriori probability of the correct word sequence given the observation sequence. Let us denote by $\mathbf{X}^1 \dots \mathbf{X}^K$, a set of training observation sequences with corresponding transcriptions $\mathbf{W}^1 \dots \mathbf{W}^K$. The MMI objective function is given by:

$$f(\lambda) = \sum_{k=1}^{K} \log \frac{P_\lambda(\mathbf{X}^k|\mathbf{W}^k)}{\sum_{\mathbf{W}} P_\lambda(\mathbf{X}^k|\mathbf{W})P(\mathbf{W})} \quad (1)$$

where λ represents the means, variances and priors of the Gaussians. The numerator of (1) represents the standard maximum likelihood objective function. In addition, MMI requires minimizing the denominator term which can be thought of as the unconditional likelihood of the data obtained by marginalizing over all the possible word sequences. The most common way to optimize (1) is the extended Baum-Welch algorithm, leading to the following update equations (i-Gaussian index, j-dimension):

$$\hat{\mu}_{ij} = \frac{\theta_{ij}^{num}(X) - \theta_{ij}^{den}(X) + D\mu_{ij}}{\theta_{ij}^{num}(X) - \theta_{ij}^{den}(X) + D} \quad (2)$$

$$\hat{\sigma}_{ij} = \frac{\theta_{ij}^{num}(X^2) - \theta_{ij}^{den}(X^2) + D(\sigma_{ij}^2 + \mu_{ij}^2)}{\theta_{ij}^{num}(X^2) - \theta_{ij}^{den}(X^2) + D} - \hat{\mu}_{ij}^2 \quad (3)$$

where $\theta_{ij}(X)$ and $\theta_{ij}(X^2)$ are sums of data and squared data weighted by mixture posterior probability as given by the Forward-Backward algorithm for either the numerator or the denominator term. The choice of the constant D is critical for the effectiveness of MMI. We refer the reader to [7] for further details about how this constant should be chosen.

One nice feature about the Aurora task is that the denominator statistics for MMI can be computed exactly. This is not the case for large vocabulary recognition where the summation over all the possible paths $\mathbf{W}$ for the denominator in (1) is intractable and has to be restricted to only the paths which occur in a word lattice. The lattice is created during a first-pass decoding step using simpler acoustic and/or language models [7].

Our approach to MMI training is to run Forward-Backward (instead of Viterbi) directly on the decoding graph from subsection 3.2 to get the posterior probability of each state at each time frame. Then, by summing across all the states that map to the same output distribution (leaf), we obtain the posterior probability of each leaf at each time frame which, when multiplied with the normalized mixture component likelihoods (or relative Gaussian activation), provides the mixture component posterior probabilities. During the Forward-Backward computation, the acoustic likelihoods are scaled down by 1/15 and the word insertion penalty is set to -1 (identical to the decoding setup). Based on the component posteriors, it is a straightforward matter to compute the denominator statistics required for the re-estimation of the means and the variances according to (2) and (3).

4.2. Probability distribution function matching

The probability distribution function matching (also called cumulative density function matching) technique was first proposed in [6] in the context of unsupervised speaker adaptation. It consists in finding a non-linear mapping for each individual dimension of the feature vectors such that the distribution function of the test data coincides with the distribution function of the training data. More formally, we define the distribution function or cumulative density function (CDF) of a continuous random variable X

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} p(t)dt \quad (4)$$

where p is the probability density function of X (if it exists). Given the training samples $x_1, \dots, x_N$, F can be approximated by the empirical CDF

$$F_N(x) = \frac{1}{N} \sum_{i=1}^{N} \theta(x - x_i) \quad (5)$$

with θ denoting the step function. The idea now is to match the empirical test CDF to the empirical training CDF for each dimension independently. Figure 1 shows the training CDF and the CDF of a particular test set for the first dimension of the LDA+MLLT features. It can be seen that there is a significant mismatch which this technique aims to minimize.

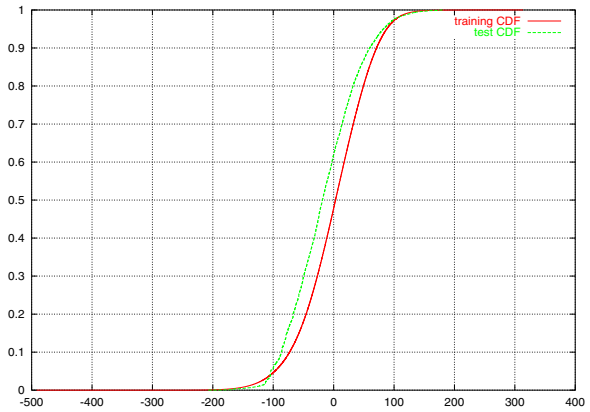

Figure 1: Example of training and test CDFs.

Let us denote by $\mathcal{T} = \{x_1, \ldots, x_N\}$ the training data for a particular dimension and by F_N the empirical training CDF, by $\mathcal{A} = \{y_1, \ldots, y_M\}$ the adaptation data and by G_M the empirical test CDF. We then can define the mapping $h : \mathcal{A} \rightarrow \mathcal{T}$, $h = F_N^{-1} \circ G_M$. From the construction of h the following property holds

$$F_N(h(y_i)) = G_M(y_i), \qquad \forall y_i \in \mathcal{A} \tag{6}$$

meaning that the training CDF value of $h(y_i)$ is the same as the test CDF value of y_i. From a practical standpoint, we first note that

$$F_N(x_i) = \frac{rank(x_i)}{N} \tag{7}$$

where $rank(x_i)$ is the rank of x_i in the sorted list of samples. The CDF matching procedure consists in a) sorting the training data b) sorting the test data and c) replacing each test sample y_i with the training sample $h(y_i)$. This is slightly different from what was proposed in [6], where the authors perform a binning of the samples and a piecewise linear warping (linear within a specific bin). After this algorithm is executed, all the test samples are replaced with training samples and one will decode training data [1]. Another observation is that the independent warping of each dimension affects the correlation of the features and therefore the CDF matching has to be followed by a decorrelating transformation such as MLLT or FMLLR. This is indicated in Table 1, where the CDF matching actually hurts accuracy although, when followed by FMLLR, it leads to a similar performance as the adapted baseline. The baseline in Table 1 has 3.3K Gaussian mixture components (comparable in size to our last year's system).

[1] Although individual samples in each dimension are part of the training data, the resulting frames are not guaranteed to lie in the training data.

5. SUMMARY

Tables 1 and 2 show the results obtained using the multi-condition models after applying the techniques described in this paper. The CDF matching performance illustrated in Table 1 turned out to be essentially the same as the baseline after adaptation. Preliminary experiments on the clean models indicate however gains in the range of 20%-30% relative suggesting that this technique may be more suitable for mismatched conditions. In addition, retraining the models on CDF-warped training data may be beneficial as was pointed out in [4]. From Table 2, our baseline accuracy has improved by 25% relative mainly due to cross-word context dependent modeling and an increase in acoustic model size. MMI training leads to an 8% relative improvement and last year's adaptation technique results in an additional 15% relative decrease in word error rate.

6. REFERENCES

[1] L.R. Bahl, P.F. Brown, P.V. de Souza and R.L. Mercer. Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition. In Proc. ICASSP'86, Tokyo, 1986.

[2] S.S. Chen and R.A. Gopinath. Model Selection in Acoustic Modeling. In Proc. Eurospeech'99, Budapest, Hungary, 1999.

[3] M. Lieb, A. Fischer. Experiments with the continuous Philips ASR system on the Aurora noisy digits database. In Proc. Eurospeech 2001, Scandinavia.

[4] S. Molau, H. Ney and M. Pitz. Histogram based normalization in the acoustic feature space. In Proc. ASRU'01, Italy, 2001.

[5] G. Saon, J. Huerta, E. E. Jan. Robust Digit Recognition in Noisy Environments: The IBM Aurora 2 System. In Proc. Eurospeech 2001, Scandinavia.

[6] S. Dharanipragada, M. Padmanabhan. A non-linear unsupervised adaptation technique for speech recognition. In Proc. ICSLP'00, Beijing, 2000.

[7] P. Woodland, D. Povey. Large Scale Discriminative Training for Speech Recognition. In ISCA ITRW Automatic Speech Recognition: Challenges for the Millenium, Paris, 2000.

[8] G. Zweig, G. Saon and F. Yvon. Arc Minimization in Finite State Decoding Graphs with Cross-Word Acoustic Context. Submitted to ICSLP'02, Denver, 2002.

System	Test	-5dB	0dB	5dB	10dB	15dB	20dB	Clean	0-20dB
Baseline	TSA	34.4	72.0	91.2	96.6	98.4	99.0	99.1	91.4
no adaptation	TSB	28.4	68.5	90.3	96.7	98.5	99.0	99.1	90.6
	TSC	34.8	72.3	90.8	96.5	98.0	98.7	99.2	91.3
FMLLR adaptation	TSA	38.9	75.8	92.5	97.6	98.7	98.9	99.1	92.7
	TSB	33.7	72.1	91.3	97.2	98.6	99.0	99.1	91.6
	TSC	40.5	76.9	92.3	96.8	98.2	98.5	99.2	92.6
CDF matching	TSA	29.1	65.9	89.2	96.6	98.1	98.5	98.0	89.7
	TSB	24.9	62.7	88.6	96.2	98.0	98.5	98.0	88.8
	TSC	27.2	64.3	88.7	95.9	97.7	98.2	98.0	89.0
CDF matching	TSA	39.5	76.0	92.9	97.6	98.6	98.8	98.9	92.8
FMLLR adaptation	TSB	33.1	71.6	91.6	97.1	98.4	99.0	98.9	91.5
	TSC	39.1	75.8	92.8	97.2	98.2	98.4	99.0	92.5

Table 1: Word recognition accuracies for the CDF matching algorithm. Systems have 3.3K Gaussians.

System	Test	-5dB	0dB	5dB	10dB	15dB	20dB	Clean	0-20dB
Last year's baseline	TSA	28.9	67.2	88.8	95.9	97.7	98.3	99.1	89.6
ML training	TSB	20.2	60.7	85.0	93.9	97.2	98.3	99.1	87.0
no adaptation	TSC	35.3	70.8	89.2	95.4	97.3	98.1	99.1	90.2
Current baseline	TSA	40.7	74.7	91.3	96.7	98.3	98.8	99.1	92.0
ML training	TSB	34.6	70.9	90.3	96.5	98.4	98.8	99.1	91.0
no adaptation	TSC	41.6	75.1	91.0	96.4	98.0	98.5	99.2	91.8
MMI training	TSA	40.9	75.9	91.7	96.9	98.5	99.0	99.3	92.4
no adaptation	TSB	36.8	73.8	91.8	97.0	98.7	99.0	99.3	92.1
	TSC	40.3	75.7	91.3	96.4	98.4	99.0	99.4	92.1
MMI training	TSA	40.3	78.3	93.8	98.0	99.0	99.2	99.5	93.7
FMLLR-P adaptation	TSB	35.0	75.2	93.0	97.9	99.0	99.3	99.5	92.9
	TSC	41.7	79.0	93.8	97.6	98.6	99.1	99.6	93.6

Table 2: Word recognition accuracies for the multi-condition system. Systems have 9.5K Gaussians.