

RECENT IMPROVEMENTS IN VOICEMAIL TRANSCRIPTION

G. Zweig, G. Saon, M. Padmanabhan, J. Huang, S. Basu
IBM T. J. Watson Research Center
P. O. Box 218, Yorktown Heights, NY 10598
Email : mukund@us.ibm.com, Phone : 914-945-2929

1 INTRODUCTION

In this paper we report recent improvements in voicemail transcription. The voicemail transcription task was introduced last year [1] as representing a style of conversational telephone speech that is somewhat different from the Switchboard and CallHome [2] databases. Last year, the speaker independent and speaker adapted word error rates (WER) on this task were reported at 41.94% and 38.18% respectively, in [1]. This year, we report a relative improvement of 18% in the speaker independent performance and 11% in the speaker adapted performance over last year. This improvement is a result of some new algorithms and an increase in the amount of training data. In the following sections, we describe the contribution of several components to improving the word error rate.

2 ACOUSTIC MODELS

2.1 Training/Test data

The starting point for the experiments reported in this paper was the system described in [1]. This system was trained on 20 hours of voicemail data (a superset of the Voicemail Corpus 1 available through the LDC - www ldc.upenn.edu), and had a speaker independent error rate of 41.94% and speaker adapted performance of 38.18%. These error rates were reported on a test set comprising of 43 voicemail messages; this will be used as the development test set for the purpose of reporting results on various algorithms in the following sections.

We have also continued our efforts to collect more voicemail training data and have succeeded in doubling the size of the database that was used last year. The training database now comprises 40 hours of speech, (400k words of text) and the size of the vocabulary has increased from 10k to 14k words.

2.2 Tree growing experiments

As the amount of voicemail training data is still limited compared to other corpora such as Hub4 and Wall Street Journal, we attempted to make use of these alternate data sources to improve the performance (experiments reported in [1] showed that using the Switchboard data did not help, possibly because of the high error rate in the training transcriptions - we plan to revisit this with the cleaned up transcriptions that are being made available now [3]). We experimented with using (i) bandlimited WSJ data (60 hours)(ii) bandlimited Hub 4 [4] data (from the F0 and F1 conditions) (40 hours) and (iii) from the Voicemail data (20 hours). Subsequently, the gaussians modelling the leaves of the tree were trained using the Voicemail acoustic data. The results are summarized in [5] and indicate that the use of the bandlimited WSJ data for constructing the trees gives the best performance.

2.3 Feature extraction experiments

Our initial experiments used 13-dimensional Mel cepstra and their first and second derivatives, but we also experimented with using alternative features such as PLP cepstra [6] and linear discriminant features. Finally, we experimented with the use of smoothed estimates for the Mel cepstra [7], the rationale being that the smoothing would lead to a reduction in the variance of the estimated feature vectors, thus leading to "tighter" models (we compute the Mel cepstra every 2 ms and average five adjacent cepstral vectors to extract one every 10 ms). The results are summarized in [5], and indicate that the best results are obtained for the smoothed Mel cepstra, which we will use in all following experiments.

2.4 Modelling dependencies through bayesian networks

Bayesian networks are a general way of representing and computing with probability distributions [8]. A distribution over a set of random variables is represented as a directed acyclic graph where each random variable X_i occurs as a node in the graph, and arcs indicate conditioning relationships. Denoting variable X_i 's predecessors in the graph by $Parents(X_i)$, the values of X_i 's parents by $Values(Parents(X_i))$, and specific variable values in lower case, the joint distribution $P(\mathbf{X})$ is factored as

$$P(\mathbf{X}) = \prod_i P(X_i = x_i | Values(Parents(X_i))).$$

We refer to variables with unknown values (i.e. about which we have no direct experimental evidence) as hidden variables, and to the others as observations. Bayesian networks have associated dynamic programming algorithms for computing the quantities that are important in speech recognition: the likeliest values for hidden variables (Viterbi decoding), the probability of an observation sequence, posterior marginal distributions over hidden variable values, and model parameters (via EM) [9, 10]. Since these algorithms work for arbitrary network structures, a Bayes-net system is convenient for rapidly testing different probabilistic models.

Table I

| Base | Linked Obs | Linked Mix | Linked Both | Cluster |
|-------|------------|------------|-------------|---------|
| 38.57 | 38.07 | 38.52 | 38.22 | 38.97 |

Bayesian networks have previously been applied in speech recognition to isolated word recognition [11], and in this section we present the first results on a continuous speech task. The networks we tested essentially incorporate a single binary-valued auxillary variable, and either the gaussian mixture weights, or the gaussians themselves or both are conditioned on this auxillary variable. These systems are referred to as 'linked mix', 'linked obs' or 'linked both' in Table I.

In our experiments, we used the acoustic score generated by a Bayes net to rescore the 100 best hypotheses generated by an 80k Gaussian system. All our models used approximately 45k Gaussians. No language model was used. We present five scores: one for the standard IBM system, and the remainder for the Bayes net system with the context variable connected to either the observation variable, the mixture component variable, or both. Finally, we present a number

for unsupervised utterance clustering. These are summarized in Table I. Although these variations are not statistically important, they indicate that the performance of our Bayes net system is at least comparable to a more standard HMM. We are currently studying the effects of varying the way in which the network is initialized, and are looking for patterns in the learned parameters.

2.5 Modelling pdf's with non-gaussian models

Purely gaussian densities have been known to be inadequate for the purpose of modelling pdf's in speech recognition systems due to the heavy tailed distributions observed by speech feature vectors. In most of the speech recognition literature, pdf's are modelled as mixtures of gaussian densities. The only attempt to model the phonetic units in speech with nongaussian mixture densities is [13], where Laplacian densities were used with a heuristic estimation algorithm.

In [12] we attempted to model speech data by building probability densities from a given univariate function $h(t)$ for $t \geq 0$. Specifically, we considered mixtures models from component densities of the form

$$p(x|\mu, \Sigma) = \rho_d \frac{1}{\sqrt{\det \Sigma}} \exp(-h(Q(x))), \quad x \in \mathcal{R}^d \quad (1)$$

where

$$Q(x) = \gamma_d (x - \mu)^t \Sigma^{-1} (x - \mu); \quad x \in \mathcal{R}^d, \quad (2)$$

$$m_\beta = \int_{\mathcal{R}_+} t^\beta f(t) dt, \quad (3)$$

$$\rho_d = \frac{\binom{d}{2}}{\pi^{\frac{d}{2}}} \frac{(m_{\frac{d}{2}})^{\frac{d}{2}}}{(m_{\frac{d}{2}-1})^{\frac{d}{2}+1}}; \quad \text{and} \quad \gamma_d = \frac{m_{\frac{d}{2}}}{dm_{\frac{d}{2}-1}}. \quad (4)$$

The vector $\mu \in \mathcal{R}^d$ and the positive definite symmetric $d \times d$ matrix Σ are respectively the mean and the covariance of this density. Particular attention was given to the choice $h(t) = t^{\alpha/2}$, $t > 0$, $\alpha > 0$; the case $\alpha = 2$ corresponds to the gaussian density, whereas the laplacian case considered in [13] corresponds to $\alpha = 1$. Smaller values of α correspond to more peaked distributions ($\alpha \rightarrow 0$ yields the δ -function), whereas larger values of α correspond to distributions with flat-tops ($\alpha \rightarrow \infty$ yields the uniform distribution over elliptical regions). For more details about these issues see [12]. This particular choice of family of densities has been studied in the literature and referred to in various ways e.g., α -stable densities as well as power exponential distributions, cf. [14]. More recently, we have also

become interested in automatically finding the ‘best’ value of α directly from the data.

Recognition experiments were carried out on the voicemail as well as the broadcast transcription task HUB4’98 by allowing different mixture components to have different values of the parameter α as compared with the fixed values $\alpha = 1$ and $\alpha = 2$. The preferred values of α tends to be less than 1.0, both for the voicemail and for the HUB4 task confirming on a systematic basis that nongaussian mixture components are preferred. An additional interesting point was that the distribution of the α values was much wider for the voicemail task than the HUB4 task. The reason for this could be the highly variable nature of the voicemail data.

| | |
|------------------------------|-------|
| Baseline (BL) | 39.7% |
| $\alpha = 1$ (20 iterations) | 38.5% |
| Prototype dependent α | 38.8% |

2.6 Extending the context dependence for observation modelling

Let us assume that we have computed a viterbi alignment on a test utterance that identifies the leaf at each time frame. The desired probability computation of a sequence of observations given a sequence of leaves is now $p(x_1 \dots x_T / l_1 \dots l_T)$, where x_t represents the feature vector at time t , and l_t represents the leaf at time t . Applying Bayes rule and ignoring the dependence of x_t on previous feature vectors, this may be written as $p(x_1 / l_1 \dots l_T) p(x_2 / l_1 \dots l_T) \dots$. Note that each term is conditioned on the entire leaf sequence $(l_1 \dots l_T)$. We normally make the approximation that all terms in this conditioning are irrelevant other than the leaf at the current time, i.e., $p(x_1 / l_1) p(x_2 / l_2) \dots$. Rather than make such an assumption, we propose to include the leaf at the previous time frame also in the conditioning term. Hence, the probability computation would be $p(x_1 / l_1) p(x_2 / l_1, l_2) p(x_3 / l_3, l_2) \dots$.

An initial implementation of this idea simply involved changing the clustering procedure by means of which the gaussians representing a leaf are constructed. The results for a system with 71k gaussians constructed using this technique is a 37.97 which compares well with a WER of 39.9386k gaussians constructed the standard way. The results look promising and we are continuing our work further in this area.

2.7 Model Complexity Adaptation

In our system, each leaf of the decision tree is modelled by a mixture of gaussians. In an earlier paper [15], we had described how to select the number of gaussians for a leaf. The essence of the algorithm is to start with a small baseline system, S1, and evaluate the probability of correct classification of the leaf in the training data. If this probability is below a threshold, t , it implies that the model for the leaf does not match the data for the leaf very well; hence, the resolution of the model for the leaf is increased by using the model for the leaf from a larger system, S2. The corresponding adapted system is referred to as S1xS2-t. The results are tabulated in Table III, and indicate that the performance of the adapted system is always somewhere between the performance of the S1 and S2 systems, and generally provides better performance for the same number of gaussians. Hence, it appears to be an efficient way of compacting a system, rather than improving on the best performance as obtained with our standard techniques.

| Old system | | | | |
|------------|--------|--------|--------|---------|
| 10 | 20 | 30 | 50 | 100 |
| 24k | 44k | 60k | 86k | 125k |
| 41.09 | 39.27 | 39.93 | 39.93 | 37.11 |
| MCA system | | | | |
| 10x100 | 20x100 | 30x100 | 50x100 | 100x150 |
| 31k | 49k | 64k | 88k | 127k |
| 39.68 | 38.47 | 38.97 | 38.67 | 37.56 |

2.8 Post-processing recognizer outputs using ROVER

In order to exploit the differences in the errors made by our systems, we used NIST’s ROVER (Recognizer Output Voting Error Reduction) [16] as a post-processor of the various word hypotheses scripts provided by these systems (presumably to take advantage of the fact that the errors made by different systems are in some sense complementary). ROVER comprises an alignment module which computes a composite word transition network (WTN) from two or more scripts by means of pairwise dynamic programming between scripts and/or WTNs. The second module scores the final WTN using one of several possible voting procedures. In the following, we will briefly describe the systems which were combined:

- The first system (BL) had 127k gaussians and $\approx 3k$ leaves and represented the baseline (10ms frame rate, decision trees use left context and

within-word right context only to predict context dependent variation of a phone).

- The second system (HF) is the equivalent of the above system, but uses a higher frame rate of 5 ms. Further, the HMM topologies were changed to preserve the same minimum duration for all phones as for the baseline. This system is used to rescore the top 100 hypotheses produced by the first system.
- The third system (RC) uses decision trees that use both left and right context across word boundaries. This system had 3017 leaves and 125k gaussians.
- The last system (SA) is a speaker adapted system described in Section 2.8.

Empty scripts are used to avoid possible insertions and to transform substitutions into deletions if all the word hypotheses at a current step in the WTN are different. As can be seen from Table IV, ROVER reduces the speaker adapted WER by an additional 3.37% (relative).

| Individual systems | |
|--------------------------------------|--------|
| Baseline (BL) | 37.01% |
| Right-context (RC) | 38.47% |
| Higher frame rate (HF) | 36.51% |
| Speaker-adapted (SA) | 33.99% |
| Rover voting | |
| Rover1 = HF + BL + RC + <i>empty</i> | 35.45% |
| SA + Rover1 + <i>empty</i> | 32.88% |

3 ADAPTATION

Most adaptation techniques generally start with speaker-independent (SI) acoustic models, and adapt them in some way [17]. We have attempted to obtain better performance by starting from models that are better matched to the test speaker than the SI model [18]. The training data is clustered into several classes, and one or many clusters that are close to the test speaker are selected and transformed independently or *jointly* to come closer to the test speaker. Subsequently, the transformed models were linearly combined so as to maximize the likelihood of the adaptation data.

For each cluster, a cluster dependent system is trained using only the speech data from this cluster and smoothed back to the SI model. When a test message is given, the cluster models are ranked according to the distances between clusters and test data, and

the closest cluster or the closest few clusters are chosen. Then, the model for each of the selected cluster(s) is transformed to bring the model closer to the test message.

We experimented primarily with linear transformations. This can be done either by the MLLR approach or the Cluster Transformation (CT). For the case where multiple clusters are chosen, the transformation for each cluster model is computed either *independently* of all the rest or *jointly*. When several cluster models are used to obtain the adapted model, it seemed to make more sense to compute the transformations of the individual cluster models jointly so as to maximize the likelihood of the adaptation data (details see [19]).

Here we only present results for the 4-cluster case (see [19] for more details). Note that the covariances of cluster-dependent models may be cluster-independent (denoted *civar*) or cluster-dependent (denoted *cdvar*). In CT the variances are cluster-independent. When multiple clusters are chosen, clusters transformations of the individual cluster means can be computed independently of one another (*cmllr-i*), or jointly (*cmllr-j*). The results are tabulated in Table V.

| Closest cluster (relative impr in ()) | | | | |
|---------------------------------------|----------------|----------------|----------------|-----------------------|
| baseline | mllr | cmllr (-civar) | cmllr (-cdvar) | ct |
| 37.97 | 35.95 (5.3) | 35.80 (5.7) | 35.15 (7.4) | 34.74 (8.5) |
| Closest two clusters | | | | |
| baseline | mllr | cmllr-i | cmllr-j | CT |
| 37.97 | 35.95 (5.3) | 36.91 (2.8) | 36.35 (4.3) | 35.80 (5.7) |

The baseline number to compare with is MLLR which gives a 5.3% relative improvement over SI in error rate. In contrast, the clustering of the training data does appear to help; CMLLR is better than MLLR by about 2.2%, and CT is about 3.4% better than MLLR.

An additional observation is that several messages in our test set were quite short: the average length was only 16 s. Hence, the amount of adaptation data for a short message is not enough to estimate the parameters for the transformations or decide reliably which cluster it belongs to. Therefore we decided to use MLLR for very short messages and CT for relatively long messages. This approach improves the WER to 33.99% - a 10.5% improvement over baseline, and a 5.5% improvement over MLLR.

4 CONCLUSION

In this paper we report recent improvements in voicemail transcription. The overall performance (word error rate) on this task has improved by 18% (relative) and 11% respectively from the performance last year, and we describe the various components that brought about this improvement. These components include experimenting with different features, generalization of HMM topologies to bayesian networks, modelling pdf's with non-gaussian models, use of a voting scheme (ROVER) to combine several hypotheses, and adaptation techniques that cluster training speakers and adapt the cluster models rather than the speaker independent models.

5 ACKNOWLEDGEMENT

We would like to acknowledge the support of DARPA under Grant MDA972-97-C-0012 for funding this work.

REFERENCES

- [1] M. Padmanabhan et al., "Transcription of new speaking styles - Voicemail", Proceedings ARPA Hub4 Workshop, Lansdowne VA, Feb 1998. *Also available at <http://www.nist.gov/speech>.*
- [2] Proceedings of LVCSR Workshop, Oct 1996, Maritime Institute of Technology.
- [3] J. Hamaker, et al., "Resegmentation and transcription of Switchboard", Proceedings of LVCSR Workshop, Sep 1998, Maritime Institute of Technology.
- [4] Proceedings of ARPA Speech and Natural Language Workshop, 1995, Morgan Kaufman Publishers.
- [5] M. Padmanabhan, et al., "Speech Recognition Performance on a new Voicemail Transcription task", Proceedings of the ICSLP 1998.
- [6] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech", Journal of the Acoustic Society of America, pp 1738-1752, April 1990.
- [7] S. Dharanipragada et al., "Techniques for capturing temporal variations in speech signals with fixed-rate processing", Proceedings of the ICSLP 1998.
- [8] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufman, 1988.
- [9] D. Heckerman. "A Tutorial on Learning with Bayesian Networks," Microsoft Technical Report MSR-TR-95-06. 1995.
- [10] G. Zweig. "Speech Recognition with Dynamic Bayesian Networks," Ph. D. Thesis, University of California at Berkeley. 1998. <http://www.cs.berkeley.edu/~zweig/>
- [11] G. Zweig and S. Russell. "Probabilistic Modeling with Bayesian Networks for ASR," ICSLP-98.
- [12] S. Basu and C.A. Micchelli, Parametric density estimation for the classification of acoustic feature vectors in speech recognition, in Nonlinear Modelling: Advanced Black-Box Techniques (Eds. J. A. K. Suykens and J. Vandewalle), pp. 87-118, Kluwer Academic Publishers, Boston 1998.
- [13] H. Ney, A. Noll, Phoneme modelling using continuous mixture densities, Proceedings of IEEE Int. Conf. on Acoustics Speech and Signal Processing, pp. 437-440, 1988
- [14] E. Gómez, M. A. Gómez-Villegas, J. M. Marin, A multivariate generalization of the power exponential family of distributions, Comm. Stat. — Theory Meth. 17(3), pp.589-600, 1998.
- [15] L. R. Bahl and M. Padmanabhan, A discriminat measure for model complexity adaptation", Proceedings of the ICASSP 1998.
- [16] J. G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER)", Proc. IEEE ASRU Workshop, pp. 347-352, Santa Barbara, 1997.
- [17] C. J. Legetter and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous density HMM's", Computer Speech and Language, vol. 9, no. 2, pp 171-186.
- [18] M. Padmanabhan, et al., "Speaker Clustering and Transformation for Speaker Adaptation", IEEE Trans. Speech and Signal Processing, Jan 1998.
- [19] J. Huang and M. Padmanabhan, "A study of adaptation techniques on a voicemail transcription task", elsewhere in the Proceedings.