Distributed Speech Recognition using Traditional and Hybrid Modeling Techniques

J. Stadermann, R. Meermeier[*]*, G. Rigoll*

University of Duisburg, Department of Computer Science
Bismarckstr. 90, 47057 Duisburg
Phone:+49-203-379-{4222, 4221},
Email: {stadermann, rigoll}@fb9-ti.uni-duisburg.de

Abstract

We compare the performance of different acoustic modeling techniques on the task of distributed speech recognition (DSR). The DSR technology is interesting for speech recognition tasks in mobile environments, where features are sent from a thin client to a server where the actual recognition is performed. The evaluation is done on the TI digits database which consists of single digits and digit-chains spoken by American-English talkers. We investigate clean speech and speech added with white noise. Our results show that new hybrid or discrete modeling techniques can outperform standard continuous systems on this task.

1. Introduction

With the rising use of intelligent mobile equipment (mobile phones, PDAs) the need for *distributed speech recognition* (DSR) is evident. In a DSR system the standard speech recognizer is divided into the *feature extraction* and the *feature classifier*. As shown in figure 1, the extracted features are transmitted over the (wireless) channel to a large server, where the classifier is implemented. In our case we always use a HMM based classifier and a mel-cepstrum based feature extraction. Features are computed for overlapping frames with a frame width of 25 ms and a frame shift of 10 ms (100 frames/s). Assuming n features per frame and b bits per feature, we have to transmit a bit rate BR of

$$\mathrm{BR} = b \cdot n \cdot 100 \frac{\mathrm{frames}}{s} \qquad (1)$$

The feature extraction produces 13 mel-cepstrum coefficients ($c_0, \ldots, c_{12}$) plus the logarithmic frame energy E and is mainly adopted from the work in [3]. Optionally it is possible to compute delta and acceleration coefficients so we obtain a feature vector with 14 and 42 coefficients per frame, respectively. The major issue in DSR is the transmission channel that possesses only a limited bandwidth but is otherwise considered ideal (appropriate channel coding can protect our data in real environments). Since the bandwidth is directly related to the bit rate [1], we keep the bit rate as channel characterization. In the following experiments two channels are investigated:

- Channel 1 allows a bit rate of 4.4 kbit/s (with channel coding and header the bit rate is 4.8 kbit/s which is half of the standard GSM bit rate for data transmissions and the desired rate for the AURORA framework [2], [3])

- Channel 2 allows a bit rate of 0.8 kbit/s; this channel is useful when using speech recognition together with other services (e.g. WAP) where only a small amount of the full bit rate is available for feature transmission

For both channels we can show that our hybrid recognition approaches are superior to standard solutions based on continuous recognizers. The following sections give a brief description of traditional continuous and discrete recognizers and a more detailed view on hybrid approaches. Section 5 presents the results with the two channels and section 6 gives a conclusion.

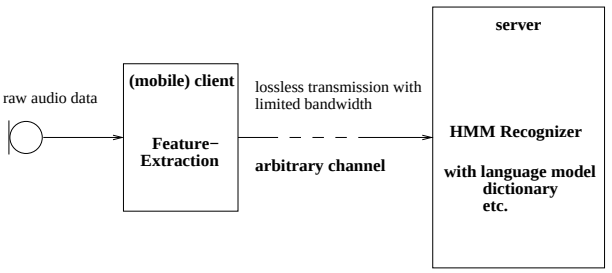


Figure 1: DSR set-up

[*]now with: SpeechWorks International, 695 Atlantic Avenue, 02111 Boston, Massachusetts, Phone: +1-617-428-4444, Email: ralf.meermeier@speechworks.com

2. Discrete hidden Markov models

The first step to create discrete HMMs is to compute a set of reference vectors (codebook) that are used to quantize the feature values from the feature extraction. For standard discrete systems we use the k-means clustering algorithm to create the codebook and quantize the features according to an Euclidean distance measure. The vector indices of the codebook vectors are then used as input for the HMMs (each HMM state has got a discrete probability value for each codebook vector index). Here, we investigate two discrete systems: The first one splits up the feature vector into 7 two-dimensional vectors, for each of this vectors the number of codebook vectors is shown in figure 2. This set-up is equivalent to the one from [3]. The

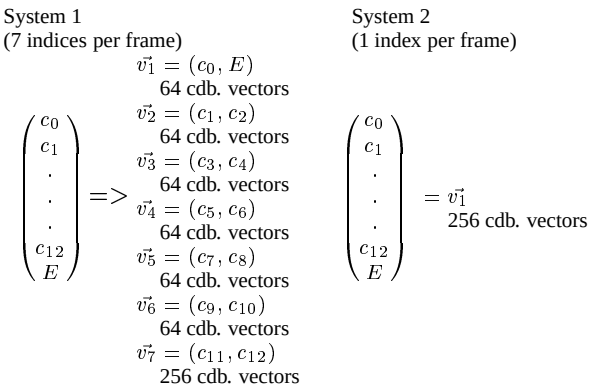


Figure 2: Codebook generation of the two discrete systems

second system contains a codebook with 256 codebook vectors for the complete feature vector (figure 2). Since the HMM recognizer only uses the vector indices we are now able to meet the bit rate limitation from the transmission channel (System 1 suited for channel 1, bit rate 4.4 kbit/s and system 2 suited for channel 2, 0.8 kbit/s).

3. Continuous hidden Markov models

This section gives a brief description of the continuous HMMs. Our system is set-up and trained according to the steps described in [3]. We use whole word hidden Markov models for the number words with 16 states each plus two silence models with 3 and 1 state, respectively. The fully trained models possess 3 Gaussian mixture densities per state (6 for the silence model) and are initialized with global mean and variance values from the training set. Assuming the transmission of the unchanged feature vector (and 4 bytes per feature value) we would get according to equation (1) a bit rate of 44.8 kbit/s or a bit rate of 134.4 kbit/s if delta and acceleration coefficients are added. The solution to reduce the bit rate is to use exactly the same quantizer that was presented in section 2. The difference on the server side is that instead of using the codebook vector indices directly in a discrete HMM,

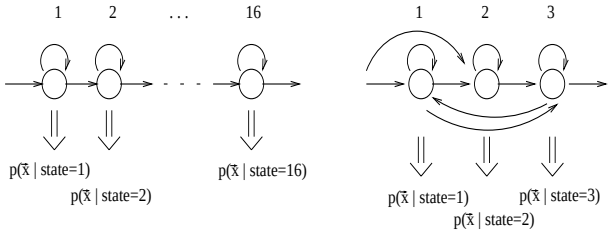


Figure 3: HMM topology for number words and 3-state silence model

we "*dequantize*" the vector indices back to continuous feature vectors by replacing them with the corresponding codebook vector (for our task the codebook can be assumed to be known on the client and on the server side).

4. Hybrid modeling techniques

The next two sections present 2 different hybrid approaches, where both extend the traditional approaches from sections 2 and 3 by inserting neural networks in the process chain.

4.1. MMI trained codebook generation

This hybrid approach tries to improve the performance of a k-means vector quantizer (VQ) by replacing the k-means VQ with a neural net (NN) and using the *maximum mutual information* between the stream of VQ indices Y and the stream of pattern classes W (e.g. phonemes or words) as objective function (for a detailed description of the idea, see [4]). This approach seems to be suited for DSR since the amount of data to be sent over the channel is unchanged compared to a standard k-means vector quantizer, but MMI-NN hybrid approaches usually outperform traditional discrete systems as shown in [4] and [5]. If we assume a parameter set Θ, the mutual information is

$$I(W, Y_\Theta) := H(Y_\Theta) - H(Y_\Theta|W) = H(W) - H(W|Y_\Theta) \quad (2)$$

The maximum of $I(W, Y_\Theta)$ can be found by minimizing $H(W|Y_\Theta)$ (with Θ as parameter):

$$\underset{\Theta}{argmin}\,(H(W|Y_\Theta)) = \quad (3)$$

$$\underset{\Theta}{argmax} \left(\sum_{k=1}^{K} \sum_{j=1}^{J} Pr(y_{j,\Theta}, w_k) \cdot \log \frac{Pr(y_{j,\Theta}, w_k)}{\sum_{\kappa=1}^{K} Pr(y_{j,\Theta}, w_\kappa)} \right)$$

This optimization is performed using a neural net with the parameter set Θ associated with the network weights. The weights are computed with a gradient descent algorithm based on (3), the details of the algorithm can be found in [5]. For our experiments we have trained the MMI-NN with the same pseudo-phonemes that are introduced in section 4.2.

4.2. Tied-posterior Markov models

The tied-posterior approach [6] can be divided into two parts:
- a *multilayer perceptron* (MLP) that exists on the client side together with the standard feature extraction. The MLP is trained with pseudo-phonemes that are derived from concatenating 4 states of the whole word models (the pseudo-phonemes of the silence models are their states themselves). Input to the MLP are feature vectors $\vec{f}$ from the current frame and from $2m$ (we chose $m = 3$) neighboring frames, output is the posterior probability of each pseudo-phoneme (index is j), $Pr(j|\vec{x})$ where $\vec{x} = (\vec{f}(t-m), \ldots, \vec{f}(t), \ldots, \vec{f}(t+m))$

- a tied-posterior (continuous) HMM recognizer on the server side that uses the *posterior probabilities* from the neural net as tied probabilities for all HMM states according to the following equation:

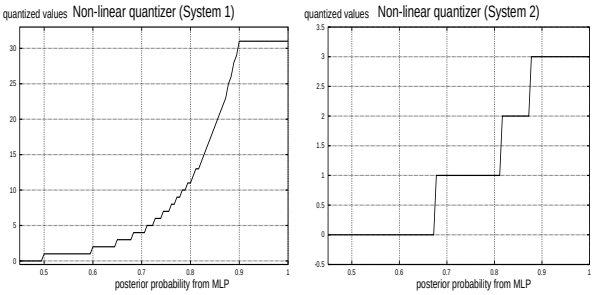
$$p(\vec{x}|S_i) = \sum_{j=1}^{J} c_{ij} \cdot \frac{Pr(j|\vec{x})p(\vec{x})}{Pr(j)} \tag{4}$$

where S_i is the HMM state, c_{ij} are the mixture coefficients and J is the number of pseudo-phonemes. Since $p(\vec{x})$ is independent of the HMM state S_i it can be omitted and (4) becomes

$$p(\vec{x}|S_i) \propto \sum_{j=1}^{J} c_{ij} \cdot \frac{Pr(j|\vec{x})}{Pr(j)} \tag{5}$$

The motivation for using this approach for DSR can be explained as follows: If we transmit in this case the quantized neuron activities instead of the quantized features, we are able to transfer a much higher information content that already includes a part of the classification information and can even accumulate the influence of multiple features (presented to the NN input layer) without the necessity of increasing the channel capacity. Since the transmission of all pseudo-phoneme-posterior probabilities takes 154 kbit/s (48 pseudo-phonemes, 32 bit per float probability value) we have (as well as in section 3) to reduce the amount of data. Fortunately the important information is stored in the top n_p probabilities (the other values are negligible) [6]. To fit in the allowed bit rate we use $n_p = 4$ for channel 1 and $n_p = 1$ for channel 2. These n_p values are then quantized using b_{np} bits and the nonlinear quantizers given in figure 4. Together with the probability values we have also to transmit its position within the array of all probabilities. For 48 MLP outputs we need 6 bit per value, so the overall bit rate for the systems is:

- System 1: 4 probability values per frame with 11 bits each (5 bit for the quantized value, 6 bit for the position) $=> BR = 4.4$ kbit/s



(a) $b_{np} = 5$ (b) $b_{np} = 2$

Figure 4: Quantizers for channel 1 and channel 2

- System 2: 1 probability value per frame with 8 bits (2 bit for the quantized value, 6 bit for the position) $=> BR = 0.8$ kbit/s

The results (section 5) prove that this approach produces the best results of all investigated methods when applying white noise and using channel 2.

5. Results

This section presents three result tables all computed on the TI digits database. The first table covers the speech-only case where the models have been trained using the clean-speech training set and were tested on the clean-speech test corpus. For table 2 and 3 the clean data from table 1 was added with *white Gaussian noise* (WGN) at a fixed signal-to-noise ratio of 6 dB and 0dB, respectively. This noisy data was again divided into training and test corpus (identical to the clean-speech corpora) and the algorithms were trained and tested only on this data. The following abbreviations are used:

- MFC42-MLP - 13 mel-cepstrum coefficients (including c_0) plus log. frame energy, with delta and acceleration coefficients, then the quantized posterior probabilities are computed

- MFC14-VQ7 - mel-cepstrum features (13 mel-cepstrum coefficients (including c_0) plus log. frame energy) quantized to seven vector indices (System 1 in figure 2)

- MFC14-VQ1 - mel-cepstrum features as above quantized to one vector index (System 2 in figure 2)

- MFC14-VQ7 MMI and MFC14-VQ1 MMI same as above, but the codebook is generated using a MMI-NN vector quantizer

- WER - word error rate

The continuous (cont.) and the tied-posterior (tied-post.) recognizers always use MFC42 on the server side.

feature extraction	recognizer	bit rate (kbit/s)	WER (%)
MFC14-VQ7	cont.	4.4	0.91
MFC14-VQ7	discrete	4.4	3.94
MFC14-VQ7 MMI	discrete	4.4	3.85
MFC14-VQ7 MMI	cont.	4.4	0.94
MFC42-MLP	tied-post.	4.4	1.86
MFC14-VQ1	cont.	0.8	5.51
MFC14-VQ1	discrete	0.8	3.78
MFC14-VQ1 MMI	discrete	0.8	3.77
MFC42-MLP	tied-post.	0.8	3.11

Table 1: Results of different acoustic modeling techniques on clean test corpus of the TI digits database

feature extraction	recognizer	bit rate (kbit/s)	WER (%)
MFC14-VQ7	cont.	4.4	4.69
MFC14-VQ7	discrete	4.4	11.98
MFC14-VQ7 MMI	discrete	4.4	11.61
MFC14-VQ7 MMI	cont.	4.4	5.26
MFC42-MLP	tied-post.	4.4	4.14
MFC14-VQ1	cont.	0.8	9.87
MFC14-VQ1	discrete	0.8	10.87
MFC14-VQ1 MMI	discrete	0.8	6.32
MFC42-MLP	tied-post.	0.8	5.23

Table 2: Results of different acoustic modeling techniques on the WGN-added test corpus ($\mathrm{SNR} = 6\mathrm{dB}$) of the TI digits database

The first column of all tables ("feature extraction") describes the feature extraction method on the client side, the second column ("recognizer") shows the type of the HMM recognizer.

For channel 2 (0.8 kbit/s), both hybrid systems always outperform the traditional approaches. The best result was achieved with the MMI-NN vector quantizer and a discrete recognizer. In case of additional white Gaussian noise, the hybrid tied-posteriors recognizer produces the best results for channel 1, a MMI approach with a continuous recognizer (with dequantized vector indices) comes second if a SNR of 0dB is set.

6. Conclusion

We have compared different acoustic modeling techniques on the task of DSR under noise-free and noisy conditions with two channel models. The results have been evaluated on the TI digits database. We are able to show that using a hybrid modeling approach (tied posteriors) reduces the word error rate by 24% relative in case of a SNR of 0dB and channel 1 compared to the best traditional result. If the bit rate is further reduced both hybrid systems (discrete MMI and tied-posteriors) achieve an error reduction of 54% and 51%, respectively (compared to the best traditional system). The next step is to use real world noise on the one hand and new feature algorithms (e.g. rasta-plp) on the other hand together with our hybrid acoustic modeling techniques.

feature extraction	recognizer	bit rate (kbit/s)	WER (%)
MFC14-VQ7	cont.	4.4	10.15
MFC14-VQ7	discrete	4.4	20.51
MFC14-VQ7 MMI	discrete	4.4	20.07
MFC14-VQ7 MMI	cont.	4.4	10.00
MFC42-MLP	tied-post.	4.4	7.75
MFC14-VQ1	cont.	0.8	28.62
MFC14-VQ1	discrete	0.8	18.00
MFC14-VQ1 MMI	discrete	0.8	8.24
MFC42-MLP	tied-post.	0.8	8.81

Table 3: Results of different acoustic modeling techniques on the WGN-added test corpus ($\mathrm{SNR} = 0\mathrm{dB}$) of the TI digits database

7. References

[1] K. David and T. Benkner, *Digitale Mobilfunksysteme*, Teubner, 1996.

[2] ETSI standard document, "Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms," in *ETSI ES 201 108 v1.1.1 (2000-02)*, 2000.

[3] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA ITRW ASR2000*, 2000.

[4] Gerhard Rigoll, "Maximum Mutual Information Neural Networks for Hybrid Connectionist-HMM Speech Recognition Systems," *IEEE Transactions on Speech and Audio Processing, Special Issue on Neural Networks for Speech*, vol. 2, no. 1, pp. 175–184, Jan. 1994.

[5] Cristoph Neukirchen and Gerhard Rigoll, "Advanced Training Methods and New Network Topologies for Hybrid MMI-Connectionist/HMM Speech Recognition Systems," in *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Munich, Apr. 1997, pp. 3257–3260.

[6] Jörg Rottland and Gerhard Rigoll, "Tied Posteriors: An Approach for Effective Introduction of Context Dependency in Hybrid NN/HMM LVCSR," in *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, June 2000.