

Distributed Speech Recognition on the WSJ task

Jan Stadermann, Gerhard Rigoll

Institute for Human-Machine Communication
Technische Universität München
Arcisstrasse 21, 80290 Munich, Germany
Phone: +49-89-289-{28319, 28541},
Email: {stadermann, rigoll}@mmk.ei.tum.de

Abstract

A comparison of traditional continuous speech recognizers with hybrid tied-posterior systems in distributed environments is presented for the first time on a challenging medium vocabulary task. We show how monophone and triphone systems are affected if speech features are sent over a wireless channel with limited bandwidth. The algorithms are evaluated on the Wall Street Journal database (WSJ0) and the results show that our monophone tied-posterior recognizer outperforms the traditional methods on this task by a dramatic reduction of the performance loss by a factor of 4 compared to non-distributed recognizers.

1. Introduction

Distributed speech recognition (DSR) is an emerging technology to implement speech recognizers on thin clients connected to a base station over a (wireless) channel. The distributed speech recognition systems that we want to investigate are based on recognizers using *hidden Markov models* (HMMs), a feature extraction module, a *Viterbi* recognition engine and a language model (figure 1). To improve the recognition performance especially in a distributed environment we present a *hybrid* speech recognizer by adding a neural net to the feature extraction module that computes posterior probabilities [1].

The idea of DSR is to let the client compute the recogni-

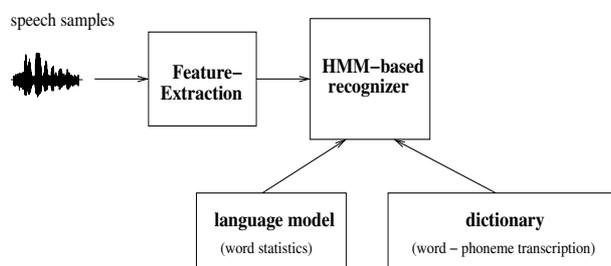


Figure 1: Standard speech recognizer architecture

tion features and then transmit these features to the base

station where a big server is located that holds the memory consuming language model and the HMMs (see figure 2). Since we consider the transmission channel to be loss-less¹ the parameter that deteriorates the recognition result (compared to a stand-alone recognizer) is the allowed bit rate on the channel.

Recently, the AURORA project [2] has defined an environment for developing front-ends for DSR. This environment defines a channel with half the bit rate of a standard GSM data transmission line (9.6 kbit/s), so the maximum bit rate is 4.8 kbit/s. Subtracting channel coding bits and header the effective bit rate usable for data is 4.4 kbit/s (more details can be found in [3]). The AURORA2 training data is based on parts of the TI digits database artificially added with noise. This database only contains single digit words or short sentences with digit chains. This paper's goal is to port our methods developed for the TI digit database and the AURORA2 database [1,4] to the WSJ0 database [5] that includes a considerably larger vocabulary (5000 words) and to demonstrate the feasibility of hybrid distributed speech recognition for a quite complex speech recognition task. Here it is no longer feasible to use whole-word HMMs as proposed in [2]. Instead we start with traditional monophones based on the LIMSI phoneme set with 47 phonemes.

Furthermore, good recognition results on this task are achieved with triphone systems, so we also want to investigate the behavior of triphone recognizers in a distributed environment. Sections 2 and 4 deal with standard acoustic models, section 5 explains the hybrid tied-posterior approach and section 6 presents the results.

2. Feature extraction and vector quantization for continuous density recognition

In this section, we first investigate the conditions required if continuous density HMMs are employed for DSR. As will be seen, a vector quantizer is needed on the client side for this scenario to transmit the features over the

¹assuming appropriate channel coding

wireless channel. The incoming audio data is sampled and divided into overlapping frames (frame width 25 ms, frame shift 10 ms). We compute 13 mel-frequency cepstrum coefficients (MFCCs) including the zeroth coefficient c_0 and the logarithmic frame energy E . The feature vector is divided into 7 sub-vectors according to the scheme proposed in [3] and depicted in figure 3. Finally, each sub-vector is quantized using a k -means vector quantizer with an Euclidean distance measure. The resulting number of vectors in each codebook is also given in figure 3. Since only the 7 codebook indices need to be sent over the channel the bit rate necessary for the transmission of the feature vectors reduces to

$$\text{BR}_{\text{VQ}} = \frac{6 \cdot 6 \text{ bits} + 8 \text{ bits}}{10 \text{ ms}} = 4.4 \text{ kbit/s}$$

3. Continuous density monophone recognizer

The continuous HMMs use Gaussian mixture probability density functions (pdf) to model the output pdf of the feature vector given the HMM state:

$$p(\vec{x}(t)|\text{state } i) = \sum_{j=1}^J c_{ij} \frac{1}{\sqrt{(2\pi)^n \sigma_{ij}^2}} e^{-\frac{(\vec{x}(t) - \vec{m}_{ij})^2}{2\sigma_{ij}^2}} \quad (1)$$

Since we only receive VQ indices from the client, we have to *decode* the data by replacing the VQ label with the corresponding codebook vector (this assumes that the codebook vectors are known on the server side).

Now, additional delta and acceleration coefficients are computed on the server side for each frame. The final feature vector contains then 42 elements (14 “restored” components plus delta coefficients plus acceleration coefficients). The HMM topology consists of 3-state HMMs for all phonemes and the sentence start/end silence model. The short pause model is a 1-state HMM. The training process applies the *Baum-Welch* re-estimation and is stopped if 12 Gaussian mixture components are reached.

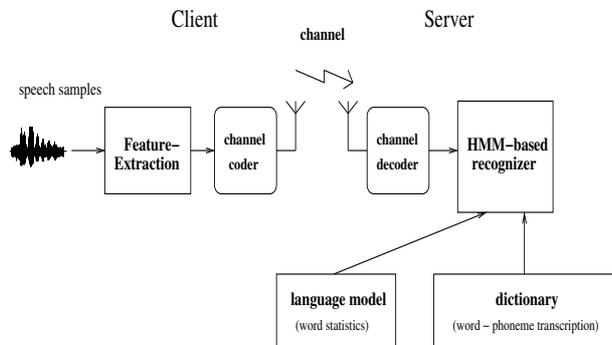


Figure 2: DSR architecture

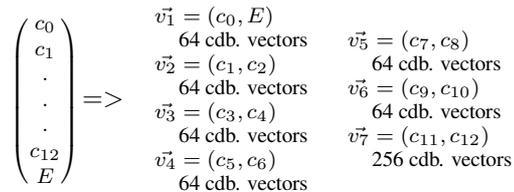


Figure 3: Codebook generation of the vector quantizer (7 codebook vector indices per frame)

4. Continuous density triphone recognizer

The triphone system uses the same features as the monophone system. Since the number of HMMs increases from 47 to 10500 we expect more distortion effects for this large number of model units, coming from the quantization error. On the other hand we gain more accuracy modeling the different speech sounds including contextual information. Experiments described in section 6 will give the answer to the question which effect will dominate the other one in a distributed recognizer. We use word-internal triphones and start with a trained single-mixture monophone system as initialization. If 6 mixtures are reached we stop the training process and start the recognition. To reduce the number of parameters in the Gaussian system we use the tree-based clustering algorithm available in the Hidden-Markov model tool kit (HTK) [6]. After tree-based clustering, the number of models is reduced to 1050.

5. Tied-posterior front-end and recognition engine

Tied-posteriors represent a hybrid NN/HMM speech recognition technology and have been introduced in [7]. It has been successfully applied to DSR in [1, 4, 8] and shall now be investigated the first time for a medium vocabulary distributed speech recognition task. The tied-posterior system is based on a *multi-layer perceptron* (MLP) that is trained to estimate phoneme posterior probabilities using the standard back-propagation training algorithm [9]. In the distributed framework, the MLP is located on the client side and the posterior probabilities are sent over the channel.

The MLP’s target values for the training process are created by a Viterbi alignment of the training set. We use the same phoneme set as in section 3, so we have 47 neural net (NN) outputs. The NN’s input layer contains the current feature vector $\vec{f}(t)$ with 12 MFCC coefficients, the logarithmic frame energy and delta and acceleration coefficients resulting in 39 components plus $2m$ adjacent feature vectors (for our experiments we chose $m = 3$). The entire input vector is then $\vec{x} = (\vec{f}(t - m), \dots, \vec{f}(t), \dots, \vec{f}(t + m))$.

The HMMs use the MLP outputs as tied probabilities for all states (see [7]). Thus, the HMM output probabilities

are given as:

$$p(\vec{x}|S_i) = \sum_{j=1}^J c_{ij} \cdot \frac{Pr(j|\vec{x})p(\vec{x})}{Pr(j)} \quad (2)$$

where S_i is the HMM state, c_{ij} are the mixture coefficients ($\sum_{j=1}^J c_{ij} = 1$) and J is the number of phonemes. Since $p(\vec{x})$ is independent of the HMM state S_i it can be omitted and (2) can be rewritten as

$$p(\vec{x}|S_i) \propto \sum_{j=1}^J c_{ij} \cdot \frac{Pr(j|\vec{x})}{Pr(j)} \quad (3)$$

Transmitting 47 probability values² would exceed the maximum bit rate so we need to quantize the NN output. Furthermore, [7] states that the important information is stored in a few NN outputs with the other NN outputs being close to zero. So we only transmit the n_p highest probabilities with b_{np} bits each using the non-linear quantizer depicted in figure 4. Additionally, the encoded index of the quantized probabilities must be known which takes $n_i = 6$ bits (47 possible positions) and the resulting bit rate is

$$BR_{TP} = \frac{n_p \cdot (b_{np} + n_i)}{10 \text{ ms}} = \frac{4 \cdot (5 + 6) \text{ bits}}{10 \text{ ms}} = 4.4 \text{ kbit/s}$$

On the server side we use the inverse quantizer to receive

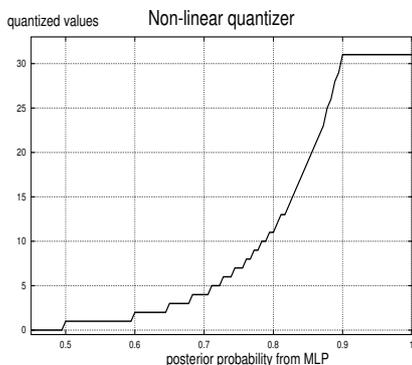


Figure 4: Non-linear quantizer ($n_p = 4$, $b_{np} = 5$)

the original values again (distorted by the quantization error). Posterior probabilities not received are set to 0.

The main advantage of the tied-posteriors approach in the DSR framework is the neural net's ability to concentrate the important class information in only a few probability values. Moreover we have information about the frame context processed in the NN and can extend this information without changing the amount of transmitted data. In [8] we have shown how this characteristic can be used to include additional features like RASTA-PLP [10] to

²stored as 4 bytes-float values

cope with non-office environments.

The triphone system using tied-posteriors is created basically in the same way as explained in section 4. The only change is the replacement of the Gaussian functions with the 47 NN outputs and the application of eq. 3. Another interesting conclusion is the fact that the difference in performance for a monophone and triphone system in distributed environments is dramatic for standard systems and very small for tied-posterior systems.

6. Results

Our results are computed on the WSJ0 database. The training set for all experiments is the official speaker independent training set si-84. Tests have been performed on the speaker independent test set si-05 with a vocabulary size of 5000 words [5]. The following abbreviations are used in the result tables:

- MFC39-MLP - 12 mel-cepstrum coefficients plus log. frame energy, with delta and acceleration coefficients, the quantized posterior probabilities are "dequantized" on the server side
- MFC14-VQ7 - mel-cepstrum features (13 mel-cepstrum coefficients (including c_0) plus log. frame energy) quantized to seven vector indices - on the server side the indices are replaced by the codebook prototypes and delta and acceleration coefficients are computed
- tied-post. - tied-posterior HMM system described in section 5
- mono - monophone HMM system
- triphone - triphone HMM system with word-internal triphones
- WER - word error rate

In both tables the first column denotes the feature extraction method used on the client side³, the second column describes the recognizer's type on the server side. The last column denotes the word error rate (WER = $1 - \text{Accuracy}$). Table 1 shows the results obtained with the Gaussian system and the tied-posterior system in normal (i.e. non-distributed) recognition mode. Table 2 presents the same systems now in the distributed environment with quantized features (bit rate 4.4 kbit/s).

First regarding the monophone systems we can observe just a slight degradation of the tied-posterior result if quantized features are used (relative loss 14%). In contrast to that the relative loss of the Gaussian system is over 65%. Thus, we can state that the tied-posterior approach is much more robust to quantization than the continuous Gaussian recognizer. The triphone systems behave a little bit different: Here, both systems are very

³in case of distributed recognition

feature extraction	recognizer	WER (%)
MFC14	cont. mono	16.64
MFC42-MLP	tied-post. mono	10.31
MFC14	cont. triphone	12.63
MFC42-MLP	tied-post. triphone	9.15

Table 1: Results on the WSJ0 test set with a standard recognizer

feature extraction	recognizer	WER (%)
MFC14-VQ7	cont. mono	27.42
MFC42-MLP	tied-post. mono	11.74
MFC14-VQ7	cont. triphone	13.37
MFC42-MLP	tied-post. triphone	9.86

Table 2: Results on the WSJ0 test set with a distributed recognizer

robust to the feature quantization (7% relative loss tied-posterior, 6% Gaussian), so we can state that the higher number of modeling units leads to more accuracy in spite of the quantization error independent of the feature transmission method.

However, in the monophone case the distributed tied-posterior system is clearly outperforming the Gaussian WSJ system, the error increase of the hybrid system is less than a quarter of the Gaussian's one. It shows that the tied-posterior architecture seems to be especially suitable for distributed speech recognition even for complex and demanding recognition tasks.

7. Conclusion

We have compared continuous and hybrid tied-posterior speech recognizers in a distributed environment where the feature extraction is separated from the other modules. The connection is established via a channel with limited bandwidth. Experiments carried out on the WSJ0 database have shown that the hybrid tied-posterior approach can cope better with quantized features than traditional continuous recognizers. The second conclusion drawn from the experiments is that a triphone system is generally much less affected by the quantization error. This effect is valid for the standard Gaussian system as well as for the hybrid one.

8. References

- [1] Jan Stadermann, Ralf Meermeier, and Gerhard Rigoll, "Distributed Speech Recognition using Traditional and Hybrid Modeling Techniques," in *Proc. EUROSPEECH*, Sept. 2001.
- [2] 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.
- [3] 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.
- [4] Jan Stadermann and Gerhard Rigoll, "Comparison of Standard and Hybrid Modeling Techniques for Distributed Speech Recognition," in *ISCA ITRW ASR2001*, Dec. 2001.
- [5] D. B. Paul and J. M. Baker, "The Design for the Wall Street Journal-based CSR Corpus," in *Proc. of the DARPA Speech and Natural Language Workshop*, Feb. 1992.
- [6] P.C. Woodland, C.J. Leggetter, J.J. Odell, V. Valtchev, and S.J. Young, "The 1994 HTK Large Vocabulary Speech Recognition System," in *Proc. ICASSP*, Detroit, Michigan, 1995, pp. 73–76.
- [7] J. Rottland and G. Rigoll, "Tied posteriors: An approach for effective introduction of context dependency in hybrid NN/HMM LVCSR," in *Proc. ICASSP*, 2000.
- [8] Jan Stadermann and Gerhard Rigoll, "Flexible Feature Extraction and HMM Design for a Hybrid Distributed Speech Recognition System in Noisy Environments," in *Proc. ICASSP*, Hongkong, China, Apr. 2003.
- [9] H. Bourlard and C. Wellekens, "Links between Markov models and multilayer perceptrons," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 12, pp. 1167–1178, Dec. 1990.
- [10] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.