Hybrid NN/HMM Acoustic Modeling Techniques for Distributed Speech Recognition

Jan Stadermann [1] Gerhard Rigoll

Technische Universität München, Institute for Human-Machine Communication, München Germany

Abstract

Distributed speech recognition (DSR) where the recognizer is split up into two parts and connected via a transmission channel offers new perspectives for improving the speech recognition performance in mobile environments. In this work, we present the integration of hybrid acoustic models using tied posteriors in a distributed environment. A comparison with standard Gaussian models is performed on the AURORA2 task and the WSJ0 task. Word-based HMMs and phoneme-based HMMs are trained for distributed and non-distributed recognition using either MFCC or RASTA-PLP features. The results show that hybrid modeling techniques can outperform standard continuous systems on this task. Especially the tied-posteriors approach is shown to be usable for DSR in a very flexible way since the client can be modified without a change at the server site and vice versa.

Key words: Distributed speech recognition, Tied-posteriors, Hybrid Speech recognition

1 Introduction

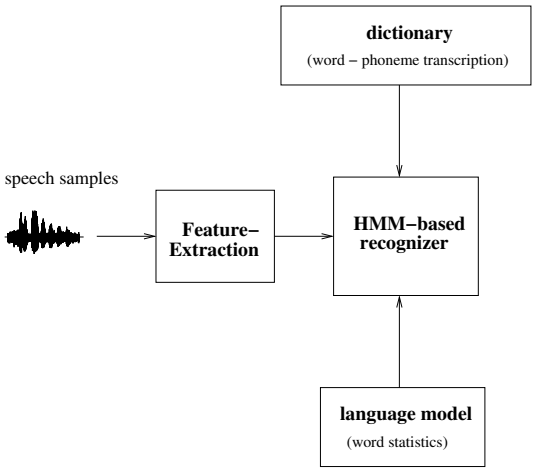
Distributed speech recognition (DSR) is a relatively new approach to integrate speech recognition technology in small, mobile devices (thin clients), such as cellular phones or personal digital assistants (PDAs). The advantage of this approach is that the speech recognition task is split up into two parts: One part that needs large memory capacities and high computation power stays on

Email addresses: stadermann@mmk.ei.tum.de (Jan Stadermann), rigoll@mmk.ei.tum.de, (Gerhard Rigoll).

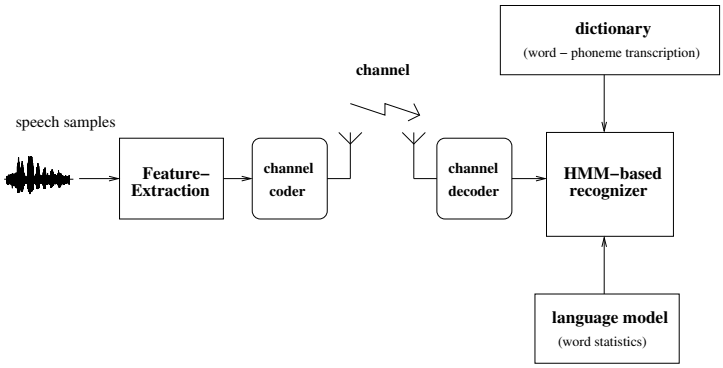
[1] now with CreaLog Software Entwicklung und Beratung GmbH, München

a big server whereas the front end with only medium requirements for storage and computation power is integrated into the client device. Both parts are connected over a (wireless) channel with limited bandwidth.
The availability of distributed large vocabulary continuous speech recognition opens a new era of human-machine communication. For instance, dictating a short message to a mobile phone instead of using the numerical keypad simplifies the SMS handling dramatically.
Other services like automatic dialogue systems could also benefit from the distribution of the speech recognition technology especially in the wireless domain because of the (already mentioned) low performance of the GSM speech coder in speech recognition scenarios. With the same reasoning multi-modal access to databases (as presented in the European project CATCH2004 [3]) or information retrieval (as presented in the European project ALERT [4]) over wireless devices could be improved because the transmission of compressed features takes much less channel capacity than the transmission of the speech signal itself. A second branch of operational areas for DSR is the integration of speech recognition in multi-modal applications such as a speech-controlled browser or a combination of touch-screen pointing and speech interaction. Here, we can only afford a very low bit rate for the speech recognition since the overall channel bit rate must be divided between the different modalities. The DSR technology is able to cope with low bit rates, especially if hybrid approaches are used.
Thus, the DSR technology is very advantegous and necessary for the above mentioned mobile scenarios. This becomes obvious if one thinks about the possible alternatives: Of course, the preferred alternative is to not separate the speech recognition system at all and transmit the audio data itself over the channel, if the channel bandwidth is large enough e.g. on a fixed telephone line (see for example [1] for speech recognition over telephone lines). Preprocessing, feature extraction and recognition is then performed entirely at the server site. But if the channel bandwidth gets narrower – e.g. on the GSM channel – the loss of information in the speech channel coder becomes too high. An investigation of speech recognition with the GSM speech coder in comparison to the recognition of MPEG coded speech data (MPEG 1, layer 3) has shown the insufficient performance of the GSM coder regarding speech recognition [2].
Another option would be to implement the entire speech recognition technology on the thin client. Progress in hardware development, especially in the area of storage and signal processor technology, has made this option a realistic technology, but we believe that for the next years, such systems implemented on the client site will be only feasible for medium vocabulary speech recognition. Especially large n-gram language models for large vocabularies (larger than 30K words) require hundreds of megabytes of memory, which could be made available on portable devices, but would make such devices much more expensive and thus would make speech recognition a relatively unattractive add-on for most users. Furthermore, it would be extremely difficult to update the lexicon or adapt the language model in such cases.

Therefore, the vision of having a very large server which could possibly handle many speech recognition requests from thin clients in parallel by receiving the appropriate feature streams over wireless channels and using complex acoustic as well as extensive language models, seems to be very attractive. The lexica and language models on such a server can be efficiently updated and special language models could be even deployed for different domains. Multi-recognition systems could be combined in order to further reduce the error rate. In order to realize such a vision of separated modules for speech recogni-



(a) standard speech recognizer modules



(b) DSR set-up

Fig. 1. Using a standard speech recognizer in DSR

tion, we have to augment current state-of-the-art speech recognition systems by adding a channel coder/decoder that converts the data to meet the channel specification while preserving as much information in the data as possible. An important question to answer is where to separate the front-end modules from the server-site speech recognizer. Figure 1 shows the basic elements of a traditional speech recognizer. From above it is known that the *hidden Markov model* (HMM) recognition engine with language model and dictionary requires a lot

more computation power and memory capacity than the feature extraction. Thus, most current approaches for distributed speech recognition are concerned with the efficient distribution of the speech recognition task between the feature extraction module on the client site and the recognition part on the server site. In this case, the crucial point is the choice of the best possible system architecture, including the computation of the most suitable features and their transmission over the wireless channel, so that the information loss of these features is as low as possible, resulting into a recognition rate that comes close to the rate obtained if the speech is fed directly into the system.
In this contribution, the authors aim to demonstrate that hybrid speech recognition methods, consisting of a combination of hidden Markov models and neural networks (NNs) have specific advantages over traditional systems, that make them especially suitable for distributed speech recognition. The major reason for this is the fact that hybrid systems allow the direct transmission of neural excitations over the wireless channel, which could be interpreted either as firing neuron streams or directly as probabilities. While traditional systems would mainly attempt to transmit features over the wireless channel, it will be shown that the neuron streams or probabilities delivered by hybrid systems can contain more information relevant for recognition than the pure features of traditional systems and additionally are easier to be transmitted over wireless channels.
The paper is organized as follows: This Section introduces the topic and shows possible scenarios for DSR, section 2 leads into the basic architecture, sections 3 and 4 present a feature vector quantization approach and the HMM topologies investigated in this work. Continuous Gaussian HMMs are presented in section 5, hybrid NN/HMM acoustic models are introduced in section 6 together with a quantization scheme for posterior probabilities. Finally section 7 deals with the evaluation of the presented models and section 8 concludes the paper.

2 General system architecture for distributed speech recognition

As already mentioned, 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. Here, the back-end consists of HMMs, language models and dictionaries. If Gaussian acoustic models are used the front-end transmits indices representing a quantized feature vector to be reconstructed at the server site. This paradigm is compared to the recently introduced tied-posteriors acoustic models [5, 6]: A class posterior estimator is added to the front-end using mel-cepstrum (MFCC) or RASTA-PLP features [7] and the

quantized probability values are sent to the server, where the recognition takes place. In this case the client is able to freely choose the features, their dimensions and the amount of context used for its classification at the cost of more computational and memory requirements. The feature extraction produces 13 mel-cepstrum coefficients $(c_0, \ldots, c_{12})$ or 9 RASTA-PLP coefficients plus the logarithmic frame energy E. Optionally it is possible to compute delta and acceleration coefficients. One major issue in DSR is the transmission channel that possesses only a limited bandwidth but is otherwise considered ideal (appropriate channel coding can protect the data in real environments). For details about dealing with lost frames or packets see [8]. The net bit rate of our channel is 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 [9, 10])

3 Vector quantization of continuous feature vectors

The base-system's feature extraction component generates 13 mel-frequency cepstral coefficients (MFCC) per frame (including the 0th coefficient). All coefficients together with the frame energy form the frame's feature vector: $\vec{f} = (c_0, c_1, \ldots, c_{12}, E)^T$. Since all coefficients are `float` values that – in general – occupy 4 bytes per value we would obtain a bit rate of

$$\mathrm{BR}_{cont} = \frac{4 \cdot 14 \cdot 8\mathrm{bits}}{10\mathrm{ms}} = 44.8\mathrm{kbit/s} \tag{1}$$

which exceeds the desired bit rate roughly by a factor of 10. One solution to reduce the bit rate is to introduce a vector quantization (VQ) process. The first option is then to use discrete HMMs at the server site which directly process the VQ-labels in order to compute state emission probabilities. In [11] it is shown that discrete acoustic models cannot compete with continuous models on the AURORA2 task. The second possibility is to convert the VQ-labels back to continuous values (see section 5) and use continuous mixture HMMs for recognition on the server.
To reduce the above computed bit rate a VQ based on the *k-means clustering* algorithm with an Euclidean distance measure is used. We adopt a quantization scheme taken from the ETSI standard [9]. This scheme composes two components from the original feature vector into a new vector that is then quantized. The quantization scheme with the number of codebook entries for each sub-vector is shown in figure 2. The encoding of the sub-vectors' indices that are then transmitted requires 6 bit for $\vec{v}_1, \ldots, \vec{v}_6$ and 8 bit for $\vec{v}_7$ resulting in a bit rate of

$$\mathrm{BR}_{VQ} = \frac{6 \cdot 6\,\mathrm{bits} + 8\,\mathrm{bits}}{10\mathrm{ms}} = 4.4\,\mathrm{kbit/s} \tag{2}$$

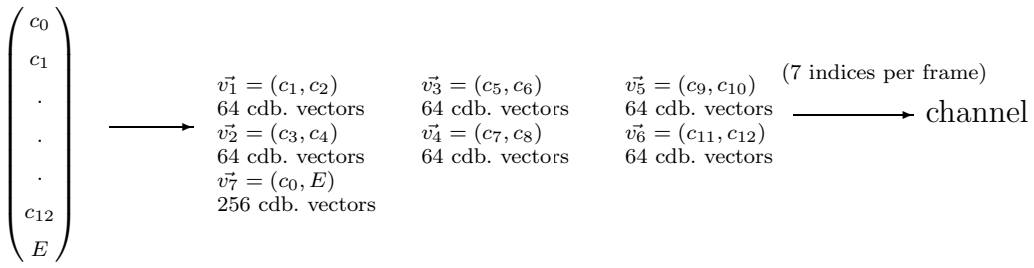


Fig. 2. Codebook generation using 7 two-dimensional VQs

4 HMM topology

For the AURORA2 recognition task (recognizing spoken digits, see section 7) we use whole word models plus two silence models for interword silence (*sp*) and sentence start/end silence (*sil*), respectively. The exact topology is depicted in figure 3. The word models for the number words one,...,nine, oh, zero have 16 states, the silence model *sil* 3 states and the *sp* model has one state [10].

Alternatively, standard phoneme HMMs with 3 states are used to model the number words (see figure 4). The complete phoneme set consists of 45 phoneme models and the two above mentioned silence models (only 20 phonemes are needed to compose the number words). Since parts of number words are very similar (e.g. o*ne*, ni*ne*) some loss in performance especially under noisy conditions is expected [12]. On the other hand, phoneme models are more flexible and larger vocabularies can be implemented easily as shown in section 7 for the WSJ0-task with a vocabulary size of 5000 words.

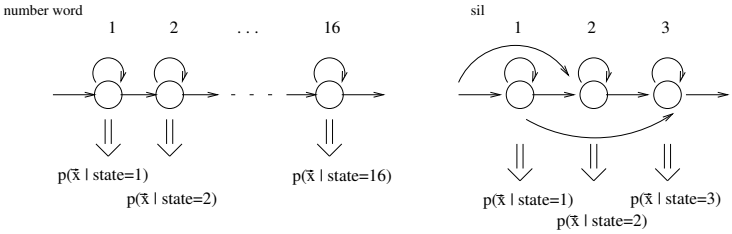


Fig. 3. Whole-word models for number words and start/end silence model

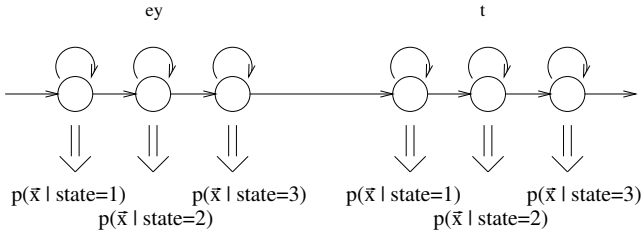


Fig. 4. Phoneme models for the number word *eight*

5 The continuous HMM system

The continuous HMM uses Gaussian mixture probability density functions (pdf) to model the output pdf of the feature vector given the HMM state. Since we receive only VQ indices from the client, we have to *decode* the data by replacing the VQ label with the corresponding codebook vector on the server (this assumes that the codebook vectors are known at the server site). Having reconstructed the continuous feature vector we can then compute additional delta and acceleration coefficients to further improve the recognition result. The final feature vector consists of 39 elements (13 "original" components plus delta coefficients plus acceleration coefficients). The 0th cepstral coefficient is discarded after re-building the feature vector.
Gaussian acoustic models with RASTA-PLP features are only built in a non-distributed environment with a base feature vector consisting of 10 components plus delta and acceleration coefficients (30 components in total). In a distributed environment a complete change of client, server and transmission scheme would be necessary since the VQ set-up from [9] does not match a RASTA feature vector with 10 components. If tied-posteriors acoustic models are used (see section 6) RASTA features can be incorporated only by changing the client. The mixture setting and training scheme for the AURORA2 task is adopted from [10]. We use 3 mixtures per state for the whole word models and 6 mixtures per state for the silence model. In case of phoneme models 5 mixtures per state are used for all HMMs trained on the AURORA2 data. The number of mixtures for HMMs trained on the WSJ0 task is stated in section 7.

6 Tied posteriors systems using neural networks

6.1 Tied-posteriors acoustic models (TP-HMM)

In recent experiments [13, 14] tied-posteriors acoustic models have proven to be superior to standard Gaussian systems in terms of flexibility and performance. We will show here that tied-posteriors are also very favorable for DSR. The tied-posteriors recognizer uses a neural network (NN) to estimate posterior probabilities $\Pr(j|\vec{x})$ for certain classes given a feature vector. Here the idea is to transmit the most important posterior probabilities over the wireless channel instead of coded feature values. Weighted sums of these posteriors are then computed in order to form state-conditional probabilities [15] at the server site. Possible neural networks are multi-layer perceptrons (MLP) or recurrent neural networks - these networks can be trained to estimate posterior probabilities [16, 17].

The classes' set-up for which the posteriors are computed is adjustable according to the given problem. Two possibilities are explored:

- Phoneme classes used for complex tasks with a large vocabulary size
- Pseudo-phonemes composed from whole-word models used for small vocabularies with a robust recognition

If a phoneme-based system is to be built it is suitable to train the NN on phoneme targets that is one neuron corresponds to the posterior probability of observing one phoneme and target values are obtained by aligning the training data (additionally this scheme can be extended to the HMM-state level [13]). In case of word-based HMMs different target values are chosen: Pseudo-phonemes are formed from the whole-word models by grouping 4 HMM states of the word model to one new unit (see figure 5). Then, one neuron computes the posterior probability of this pseudo-phoneme. This specialized pseudo-phoneme-NN has been created to do a fair comparison with the word-based Gaussian systems. The tied-posteriors approach allows to combine phoneme-based NNs with word-based HMMs as well, but only with a noticeable loss in performance compared to the above mentioned approach. Thus, we have a resulting number of 48 pseudo-phonemes ($4 \cdot 11$ units from the whole word models plus 4 from the silence models, where each state is handled separately). The input layer of the neural net consists of the feature

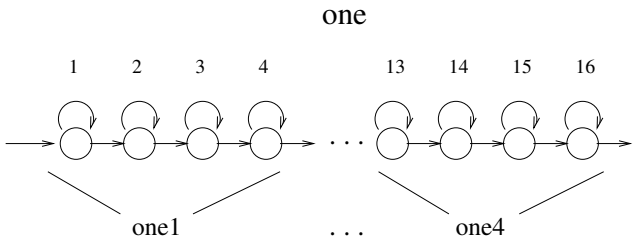


Fig. 5. Composing pseudo-phonemes from whole word HMM "one"

vector $\vec{f}$ with delta and acceleration coefficients computed on the client site from the current frame t. The MLP extends the frame's time context by $2m$ neighboring frames (in our experiments $m = 3$) obtaining an input vector $\vec{x} = (\vec{f}(t-m), \ldots, \vec{f}(t), \ldots, \vec{f}(t+m))^T$. With the feature vectors described in section 5 the NN's input layer consists of 273 (MFCC) and 210 (RASTA) nodes, respectively. The MLP's hidden layer varies according to the task between 500 nodes (AURORA2 test) and 1000 nodes (WSJ0 test).
On the server site we have a *tied-posteriors* HMM recognizer that uses the NN's outputs as tied probabilities for all HMM states. The equation of the HMM output density function is derived as follows: We first express the HMM output pdf as a sum of mixture densities:

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

where S_i is the HMM state, c_{ij} are the mixture coefficients and J is the number of classes (phonemes, pseudo-phonemes or HMM states). The probability density $p(\vec{x}|j)$ can now be expressed by the posterior probability $\Pr(j|\vec{x})$ that is retrieved from the MLP by using Bayes' rule:

$$p(\vec{x}|j) = \frac{\Pr(j|\vec{x})p(\vec{x})}{\Pr(j)} \tag{4}$$

Hence we obtain

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

Since $p(\vec{x})$ is independent of the HMM state S_i it can be omitted and (5) becomes

$$p(\vec{x}|S_i) \propto \sum_{j=1}^{J} c_{ij} \cdot \frac{\Pr(j|\vec{x})}{\Pr(j)} \tag{6}$$

The posterior probability $\Pr(j|\vec{x})$ is estimated by the NN, the a priori probability $\Pr(j)$ is approximated by the classes' relative frequencies in the training data and the mixture weights c_{ij} are trained using the standard Baum-Welch algorithm.

6.2 Quantizing posterior probabilities

Transmitting all `float` probability values over the channel for each frame would by far exceed the allowed bit rate. Fortunately it is sufficient to know the n_p highest probabilities and skip the other ones. In [15] it is reported that only a few output classes have high probabilities while the other ones are negligible.
The motivation for using this approach for DSR can be explained as follows: If we transmit 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). The amount of transmitted data is independent of the input layer size. So we can add more context frames or introduce another feature extraction method (e.g. RASTA-PLP features to better cope with noisy data [14]) by simply extending the input layer without changing the output layer size. An overview of the client site is given in figure 6. From this figure it can be seen that the amount of data to be transmitted over the channel is only dependent on the number of probabilities n_p that are selected for transmission. If the input to the quantizer is specified (e.g. using phoneme posterior probabilities) arbitrary feature vectors, classifiers and arbitrary HMM topologies are possible without changing the bit rate on the channel. To meet the bit rate requirement each of the n_p probability values is quantized with the characteristic curve $[a \cdot \exp(bx - c)]$

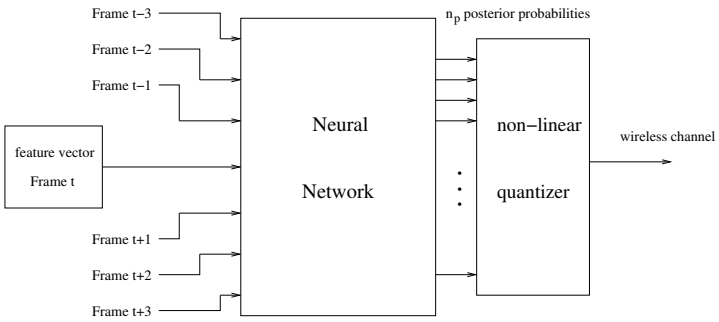


Fig. 6. Client site of a distributed tied-posterior recognizer

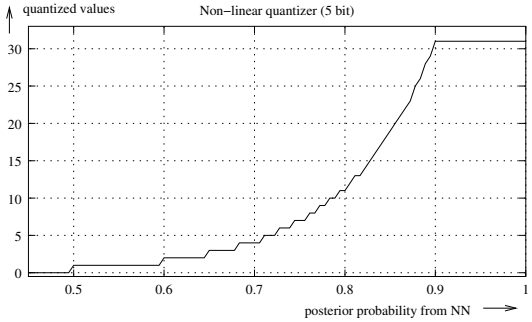


Fig. 7. Quantizer with $b_{np} = 5$

([.] denotes the ceiling function, a and b are user-definable parameters). The curve used in this work is presented in figure 7. Based on the assumption that only a few output neurons produce a high probability value the parameters a and b are set empirically. With this quantizer the error in equation (6) caused by the quantization is kept small to keep as much information as possible in the likelihood computation. Transmitting the $n_p = 4$ highest probabilities quantized with $b_{np} = 5$ bits each (plus 6 bits for the probability's index) results in the required ETSI bit rate:

$$\mathrm{BR}_{\mathrm{TPq}} = \frac{4 \cdot (5+6)\, bits}{10\,\mathrm{ms}} = 4.4\,\mathrm{kbit/s} \tag{7}$$

State-dependent likelihoods are calculated using eq. (6), with the tied-posterior weights c_{ij} from the HMMs and the received and reconstruced probability values $\hat{\Pr}(j|\vec{x})$. The recognition is performed using the resulting likelihood values. Figure 8 shows the reconstruction of probabilities at the server site. It should be pointed out here, that the standard hybrid posterior approach from [18] cannot be used in a similar manner for DSR as described above. In this classic hybrid approach the posterior probability $\Pr(i|\vec{x})$ is directly connected to the HMM state i. The number of HMMs is then identical to the NN's output layer size. Since only $n_p = 4$ probabilities are transmitted, a lot of HMM emissions are 0 which leads to disastrous results (see section 7). In contrast to that, the tied-posterior algorithm can compute all HMM densities even if only n_p probabilities are received because of additional use of the

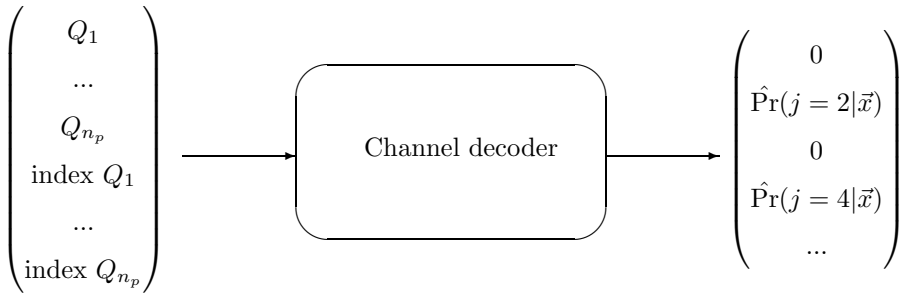


Fig. 8. Probabilities received at the server site using a distributed hybrid acoustic models

weights c_{ij} to compute these values. Therefore, the tied-posterior approach is an ideal hybrid paradigm for DSR.

7 Results

The evaluation of the presented algorithms is performed on the AURORA2 database [10]. This database contains a subset of sentences taken from the TI digits database with additive real noise at different SNRs. The samples are filtered to simulate various channel characteristics and are downsampled to a frequency of 8 kHz. Example noise types are airport noise, babble noise or car noise. The test sets contain known and unknown noise types at SNR values ranging from 20 dB to 0 dB. Test set A includes the same noise types during training and testing, test set B contains unknown noise and test C consists of a different channel with known and unknown noise types [10]. There are two different training sets: One consists of clean speech only whereas the other one contains various noise types as well as clean speech (multi-condition training). The word error rates of the evaluation (WER = 1−Accuracy) are mean values for the different SNRs (0 dB, 5 dB, 10 dB, 15 dB, 20 dB). Here, each acoustic model is trained using the multi-condition training set.
Additionally, the WSJ0 database [19] is used to evaluate DSR on a task with a larger vocabulary size. Gaussian acoustic models and TP-HMMs are trained on 7240 sentences of the speaker independent training set *si-84*. The WSJ0 speaker independent test set *si-05* has a closed vocabulary of 5000 words. The WER is computed using the test set's bigram language model with 47 context independent HMMs and context dependent triphones, respectively. Since the WSJ test consists of clean data, only MFCC features have been calculated in this case.

Table 1 presents the AURORA2 reference results (*AURORA2 ref.*) together with our standard Gaussian acoustic models in a non-distributed environment. Since the AURORA2 evaluation in [10] does not contain phoneme baseline models we take our Gaussian MFCC models (*MFC42 Gauss ref.*) as the baseline result for further experiments with phoneme models. Results marked in

client	recognizer	Test A	Test B	Test C	mean
AURORA2 ref.	whole-word	12.18%	13.73%	16.22%	13.61%
MFC42 Gauss	whole-word	**11.63%**	14.07%	16.55%	**13.59%**
RASTA30 Gauss	whole-word	13.44%	14.79%	**14.73%**	14.24%
MFC42 Gauss ref.	phoneme	16.69%	25.23%	21.94%	21.16%*
RASTA30 Gauss	phoneme	18.83%	**20.96%**	**21.18%**	**19.94%**

Table 1
Results (WER) on the AURORA2 test sets using Gauss models (multicondition training)

client	recognizer	Test A	Test B	Test C	mean
MFC42 Gauss	whole-word	13.31%	15.56%	18.02%	15.15%
MFC42 Gauss	phoneme	16.74%	**22.63%**	23.36%	**20.42%***

Table 2
Results (WER) on the AURORA2 test sets using Gauss models (multicondition training), bit rate 4.4 kbit/s

bold types denote an improvement over the baseline results.
There are two main conclusions from table 1: The Gaussian phoneme models perform worse than the word models and RASTA-PLP features are well suited particularly for channel distortions (test C) as expected. Table 2 shows the results with the same Gaussian MFCC models, but in a distributed environment with quantized features according to section 3. Interestingly the distributed recognizer with phoneme models outperforms the baseline (marked with '$*$'). Since the RASTA feature vector's size is different from the MFCC feature vector the VQ from section 3 (needed for distributed recognition) is not applicable to the Gaussian HMMs based on RASTA features.

The results in the next two tables 3 and 4 are computed using the tied-posteriors (TP) acoustic model introduced in section 6. The MFCC TP model outperforms the Gaussian reference in test A (known noise) even in a distributed system since the noise can be learned in the net weights. Changing the client to RASTA-PLP features (no changes to the quantization scheme are necessary) results in an overall better system compared to the reference. Here, the (unknown) noise and (unknown) channel are compensated by the more suitable feature extraction. These statements are true for whole-word models as well as phoneme HMMs. In contrast to Gaussian models, the advantage of RASTA-PLP features can be used in a distributed TP system without changing the transmission or the server.

client	recognizer	Test A	Test B	Test C	mean
MFC42 TP	whole-word	**8.96%**	19.40%*	22.45%*	15.83%*
MFC42 FP	whole-word	**9.21%**	19,21%	22.45%	15.86%
RASTA30 TP	whole-word	**9.29%**	**12.92%**	**11.27%**	**11.14%**
RASTA30 FP	whole-word	**9.24%**	**13.27%**	**11.22%**	**11.25%**
MFC42 TP	phoneme	**12.98%**	**24.61%**	26.70%	**20.38%**
MFC42 FP	phoneme	**13.53%**	25.16%	28.00%	21.07%
RASTA30 TP	phoneme	**13.14%**	**19.58%**	**15.59%**	**16.20%**
RASTA30 FP	phoneme	**13.32%**	**19.84%**	**15.67%**	**16.40%**

Table 3
Results (WER) on the AURORA2 test sets using TP models (multicondition training)

client	recognizer	Test A	Test B	Test C	mean
MFC42 TP	whole-word	**9.34%**	16.87%*	20.61%*	14.61%*
MFC42 FP	whole-word	97.32%	99.61%	99.57%	98.69%
RASTA30 TP	whole-word	**9.80%**	**12.89%**	**11.96%**	**11.47%**
RASTA30 FP	whole-word	83.09%	95.29%	90.88%	89.53%
MFC42 TP	phoneme	**14.26%**	25.87%	27.41%	21.53%
MFC42 FP	phoneme	78.62%	96.11%	96.16%	89.12%
RASTA30 TP	phoneme	**14.47%**	**21.25%**	**16.75%**	**17.64%**
RASTA30 FP	phoneme	49.98%	63.25%	56.15%	56.52%

Table 4
Results (WER) on the AURORA2 test sets using TP models (multicondition training), bit rate 4.4 kbit/s

A degradation of the DSR performance is observable compared to the non-distributed case except for the MFCC word models (marked with '$*$'). Additionally, a hybrid system with fixed connections (FP - fixed posteriors) between NN and HMM according to [18] is evaluated. The only difference to the TP models is the absence of adjustable mixture coefficients i.e. the likelihood of one HMM state is computed using only one neuron ($c_{jj} = 1$, all other c's are zero, see eq. 6). A slight degradation to the TP models is observable in the non-distributed case, but in a DSR system the FP models are not usable. The DSR decoder is not able to find a valid path through the models (pruning is deactivated) since the majority of HMM emission likelihoods is zero. Summa-

rizing the results from TP-HMMs and Gaussian HMMs one can state that the tied-posteriors approach is significantly better than the Gaussian one both in terms of flexibility and performance. Table 5 compares the results of different

Emission density computation	**HMM System**	**WER**
Gauss - 10 mixtures	Mono47	15,28%
Gauss - 12 mixtures	Mono47	14,78%
Gauss - 6 mixtures	Tri8379	12,54%
Gauss - 12 mixtures	Tri8379	13,88%
Tied posteriors (TP) - 47 outputs	Mono47	**10,20%**
Tied posteriors (TP) - 47 outputs	Tri10534	**9,02%**
Fixed Posteriors (FP) - 47 outputs	Mono47	90%

Table 5
Results on the *si-05* test set of the WSJ0 database, bit rate 4,4 kbit/s

acoustic models on the WSJ0 task in a distributed set-up. To emphasize the flexibility of the TP models, the phoneme-based TP-HMMs at the server site are used for the AURORA2 evaluation as well as the WSJ test regardless of the client (the NN on the client has been changed according to the given task). The first four rows of table 5 show the results of continuous Gaussian-HMMs with the given number of mixtures in each state. The basic HMM set consists of the same 45 phoneme HMMs and two silence models as described in section 4. Based on this set, 10534 triphone models are created that are clustered to 8379 HMMs (30972 mixtures in total) in the Gaussian case. The best result is obtained with triphone models and 6 mixtures per state, increasing the number of the triphones' mixtures increases the WER due to insufficient training data.
At the server site either 47 monophone HMMs or 10534 triphones are used, presented in the second half of table 5. The best result is obtained with a NN trained on the 47 monophones and a triphone-based server. Finally the system with a fixed connection between NN outputs and HMM states according to [18] is given in the last row of table 5. Again, most of the state likelihoods are zero (see section 6.2) and no hypothesis survived during the decoding process in most sentences.

8 Conclusion

Using TP acoustic models for distributed speech recognition is shown to be an interesting alternative to standard Gaussian HMMs. The TP-HMMs perform

generally better than the Gaussian HMMs and allow a very flexible adaptation to the desired task and condition. The client site can be modified, e.g. by using different features without touching the server. Apart from task-dependent word models we have also evaluated task independent phoneme models on the AURORA2 task (small vocabulary, different background noise conditions) and the WSJ0 task (medium vocabulary, no background noise). Again TP systems are superior and client and server of distributed TP recognizers can be modified easily as long as the amount of data transmitted over the channel is unchanged. Examples of these modifications include context dependent models at the server site and NNs with different input features on the client. The results show that the tied-posterior approach represents the most effective hybrid method for DSR in terms of performance and flexibility. An interesting phenomenon that has been observed is the fact that quantized MFCC features perform better on the AURORA2 task under unknown conditions (test B) than regular ones.

List of Figures

List of Tables

References

[1] D. Yuk and J. Flanagan, "Telephone speech recognition using neural networks and hidden markov models," in *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.

[2] C. Barras, L. Lamel, and J. Gauvain, "Automatic transcription of compressed broadcast audio," in *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City, Utah, USA, 2001.

[3] Henrik Schulz, Marion Mast, Thomas Ross, Heli Harrikari, Jan Stadermann, Vasiliki Demesticha, Yannis Vamvakoulas, and Lazaros Polymenakos, "A Conversational Natural Language Understanding Information System for Multiple Languages," in *6th International Workshop on Applications of Natural Language for Information Systems*, Madrid, Spain, June 2001.

[4] *Alert system for selective dissemination of multimedia information*, http://alert.uni-duisburg.de.

[5] 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.

[6] Jan Stadermann, Ralf Meermeier, and Gerhard Rigoll, "Distributed Speech Recognition using Traditional and Hybrid Modeling Techniques," in *European Conference on Speech Communication and Technology*, Aalborg, Denmark, Sept. 2001.

[7] Hynek Hermansky and Nelson Morgan, "RASTA Processing of Speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[8] Alastair James, Ángel Gómez, and Ben Milner, "A Comparison of Packet Loss Compensation Methods and Interleaving for Speech Recognition in Burst-Like Packet Loss," in *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, USA, Mar. 2005.

[9] "Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms," in *ETSI ES 201 108 v1.1.3 (2003-09)*, Sept. 2003.

[10] 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.

[11] Jan Stadermann and Gerhard Rigoll, "Comparison of Standard and Hybrid Modeling Techniques for Distributed Speech Recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Madonna di Campiglio Trento, Italy, Dec. 2001.

[12] Jan Stadermann and Gerhard Rigoll, "Verteilte Spracherkennung mit hybriden akustischen Modellen," in *31. Deutsche Jahrestagung für Akustik, DAGA05*, München, Deutschland, Mar. 2005.

[13] Jan Stadermann and Gerhard Rigoll, "Comparing NN Paradigms in Hybrid NN/HMM Speech Recognition using Tied Posteriors," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, St. Thomas, U.S. Virgin Islands, Nov. 2003.

[14] Jan Stadermann and Gerhard Rigoll, "Flexible Feature Extraction and HMM Design for a Hybrid Distributed Speech Recognition System in Noisy Environments," in *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hongkong, China, Apr. 2003.

[15] 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, 2000.

[16] Herve Bourlard and Christian Wellekens, "Links between Markov models and multilayer perceptrons," *IEEE Transactions on Pattern Analalysis Machine Intelligence*, vol. 12, pp. 1167–1178, Dec. 1990.

[17] S. Santini and A. Del Bimbo, "Recurrent neural networks can be trained to be maximum A posteriori probability classifiers," *Neural Networks*, vol. 8, no. 1, pp. 25–29, 1995.

[18] Herve Bourlard and Nelson Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[19] Douglas B. Paul and Janet M. Baker, "The Design for the Wall Street Journal-based CSR Corpus," in *International Conference on Spoken Language Processing (ICSLP)*, Banff, Canada, Oct. 1992, pp. 899–902.