FLEXIBLE FEATURE EXTRACTION AND HMM DESIGN FOR A HYBRID DISTRIBUTED SPEECH RECOGNITION SYSTEM IN NOISY ENVIRONMENTS

Jan Stadermann, Gerhard Rigoll

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

ABSTRACT

Using the client device of a distributed speech recognizer usually implies the presence of background noise since most scenarios for *distributed speech recognition* (DSR) are situated in a non-office environment. Thus, the general task is to choose the most suitable feature extraction method for the given conditions. We present a hybrid speech recognition approach implemented for DSR that allows the choice of arbitrary feature vectors (regarding number and range of value) without changing the amount of data sent to the recognition engine. Experiments were carried out using mel-cepstrum and RASTA-PLP features on the AURORA database. Results show how the recognition performance under different noise conditions can be adjusted if the different features are combined, and that our hybrid approach to DSR has advantages that could not that easily be obtained with traditional DSR architectures.

1. INTRODUCTION

In a distributed speech recognition system (DSR), the recognition engine with language model, dictionary and acoustic models is separated from the feature extraction. The connection between the client side where the feature extraction is located and the server side (with the recognition engine) is a channel with limited transmission capacity. The advantage of this approach (compared to integrating the recognizer on the client device itself) is the availability of computation power and memory resources on the server side. The recognition engine can calculate its results with much more accuracy and precision than the one running on a client device. Examples of channels are fixed networks or the air interface in mobile communication. To deal with the limited capacity a quantization of the transmitted data is necessary. The maximum channel data rate assumed here is 4.4 kbit/s, this data rate resembles to the one specified in the ETSI standard [1] for DSR front-ends ([1] allows 4.8 kbit/s inclusive overhead, the raw data rate is 4.4 kbit/s).
This paper's focus is the acoustic modeling for quantized data, therefore we assume no information loss during the transmission over the channel. In a standard[1] distributed HMM-based recognition system a vector quantizer (VQ) is necessary to compress the feature vector [2]. A major disadvantage of this standard approach is that the VQ must be designed for a fixed number and a fixed type of features. Furthermore, increasing the feature vector's dimension also increases the quantization error of the new VQ since the size of the VQ codebook is determined by the channel's data rate. Therefore, we proposed in [3] to adapt our hybrid tied-posterior algorithm to the DSR task. The hybrid recognizer can easily include context in the recognition process, it is trained in a discriminative way and the client side can be adapted to specific conditions (e.g. by using a neural net with features especially suited for noisy environments or by adding delta features) without changing the server side. On the other hand, the server side can (as well as the standard HMM systems) make use of context-dependent models (e.g. triphones) if the task requires it.
In the following section we first present the hybrid architecture suitable for DSR, then in section 3 we describe the composition of the different feature vectors and after the experimental results given in section 4 we draw a conclusion.

2. HYBRID ARCHITECTURE

The main feature of the hybrid architecture that should be explored here is the neural network's (NN) ability to estimate posterior probabilities. If the estimated probabilities are close to the real ones, the models on the server side are not dependent on a specific NN, the only requirement is that the models have been trained with the same set of posterior probabilities. Furthermore, we are using tied posterior probabilities i.e. the NN is connected with the HMMs

[1] acoustic models based on Gaussian mixture densities

From: ICASSP 2003, IEEE

via mixture coefficients. This architecture allows to adapt the HMM topology to the given task (whole word models, n-state monophones, triphones. etc.) and to transmit incomplete NN outputs from the client to the server and to restore a valid HMM output on the server side

Figure 1 shows our general set-up for DSR. Denoting

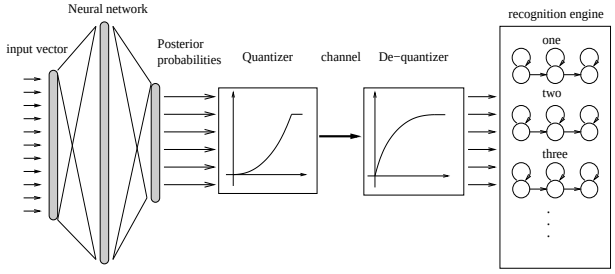

Fig. 1. Hybrid ASR architecture for distributed recognition

the feature vector of one time frame with $\vec{f}$, the first step is to introduce some context to the NN's input layer by adding feature vectors of past and future time frames: $\vec{x} = (\vec{f}(t-m), \ldots, \vec{f}(t), \ldots, \vec{f}(t+m))^T$ A second step could be to expand the feature vectors themselves with context by adding delta and acceleration values. Finally, a second set of features can be added. Denoting the two feature vectors with $\vec{f}_1$ and $\vec{f}_2$, the complete input layer then results in

$$\vec{x} = (\vec{f}_1(t-m), \vec{f}_2(t-m), \ldots, \vec{f}_1(t), \vec{f}_2(t), \ldots, \vec{f}_1(t+m), \vec{f}_2(t+m))^T$$

The NN's output layer dimension can be chosen independently from the input layer dimension. The network topology suitable for speech recognition is a fully-connected multi-layer perceptron with one hidden layer [4]. The weights are trained with the back-propagation algorithm optimizing the training set's cross entropy, for the output nodes we use the softmax function as non-linearity, the hidden nodes apply the sigmoid function.

To obtain a similar range of value at all input nodes, a normalization process takes place:

$$x_i^{(n)} = \frac{x_i - \overline{x}_i}{\sqrt{\sigma_{x_i}^2}} \tag{1}$$

$\overline{x}_i$ is the global mean value for input node i and $\sigma_{x_i}^2$ is the global variance for this node.

The NN is normally trained to calculate phoneme probabilities. One output node computes the posterior probability $Pr(N_j|\vec{x})$ that phoneme N_j has been observed given the (normalized) NN input vector $\vec{x}$. The AURORA framework does not offer a phoneme alignment, but uses whole word models [2] with 16 HMM states per model. So, pseudo-phonemes are created by concatenating 4 states of the whole

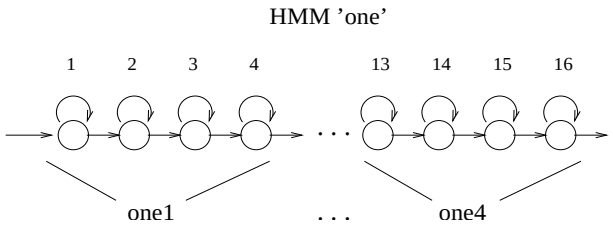

Fig. 2. Composing pseudo-phonemes from whole word HMM "one"

word models. Adding the HMM states of the silence models *sil* and *sp* the output layer size is 48 (4 nodes from the eleven digit words each plus 4 silence model nodes). The NN's output vector $\vec{y}$ is then quantized using the non-linear quantizer depicted in figure 3. The characteristic curve of the quan-

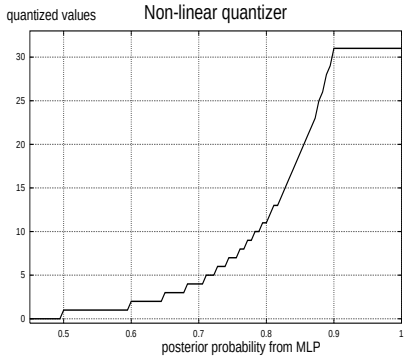

Fig. 3. Non-linear quantizer ($n_p = 4$, $b_{np} = 5$)

tizer is based on the exponential function[2] $[a \cdot \exp(by - c)]$ with clipping at the upper range of values. The selectable parameters a, b and c were adjusted by a trial-and-error method on the training set.

To meet the bit rate specified in the AURORA framework, the $n_p = 4$ highest values of $\vec{y}$ are quantized with $b_{np} = 5$ bits each. Additionally, 6 bits are necessary to encode the value's index. Assuming a frame shift of 10 ms the resulting bit rate is

$$\mathrm{BR}_{\mathrm{TPq}} = \frac{4 \cdot 11\,\mathrm{bits}}{10\,\mathrm{ms}} = 4.4\,\mathrm{kbit/s} \tag{2}$$

On the server side the quantized values are restored to a vector with zeros filled in where no component was transmitted. The probability density value needed for the output of HMM state S_i is computed according to the tied-posterior approach in [5]

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

[2] [.] denotes the ceiling function

where $Pr^{(q)}(j|\vec{x})$ is the de-quantized posterior probability sent over the channel or zero if the index j was not sent and c_{ij} is the mixture coefficient connecting the received posterior value $Pr^{(q)}(j|\vec{x})$ with the HMM state S_i. The *a-priori* probabilities $Pr(j)$ are known in advance and can be computed from the training data.

3. FEATURES

As mentioned in the introduction the major goal of this work is the exploitation of our hybrid DSR-architecture concerning its capability to allow a flexible composition of different feature sets in order to make the system more robust against changing noise conditions. The features that will be explored in the following paragraphs are all frame based. Our system creates a frame every 10 ms, the width of one frame is 32 ms. If the features are combined, it is required that frame width and frame shift are identical for all extraction algorithms.

3.1. Mel-frequency cepstrum coefficients

The mel-frequency cepstrum coefficients (MFCC) are computed in a common fashion with a mel filterbank containing 23 filters, then the first 12 cepstrum coefficients are calculated from the mel-spectrum (excluding c_0). Finally, the frame energy is appended, resulting in a base feature vector with 13 components. In the hybrid framework, delta and acceleration coefficients can be computed on the client side, the feature vector dimension has no influence on the NN's output dimension. In case of the combination with RASTA-PLP features we use only 9 cepstrum coefficients $c_1, \ldots c_9$ and the frame energy.

3.2. RASTA-PLP

RASTA-PLP features are an extension of the PLP features introduced by [6]. The RASTA filterbank modifies the PLP features and aims at suppressing slowly varying spectral distortions. To cope with additive noise components as well, [7] proposes J-RASTA features, which we used with a fixed J in our experiments. The whole process of creating the features is depicted in figure 4. The critical band spectrum

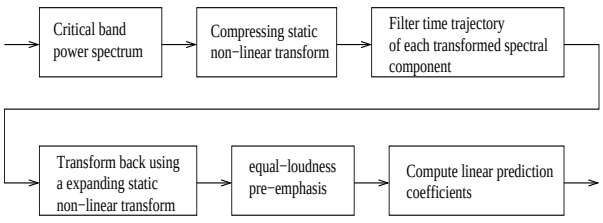

Fig. 4. Processing steps for RASTA features

filterbank contains 16 filters, the parameter J in the non-linear compression is chosen to $J = 10^{-6}$ and we insert the first 9 linear prediction coefficients in the feature vector.

4. EXPERIMENTS

The AURORA 2 database was presented in [2]. It contains a sub set of spoken digits and digit chains taken from the TI digits database added with different noise types at different signal-to-noise ratios (SNR). For our experiments we use only the multi-condition training set and compute feature vectors containing MFCC features, RASTA-PLP features and a combined feature vector using MFCC and RASTA-PLP. One HMM and one neural net are trained for each one of the three feature sets, the recognition is done for all 9 possible combinations of neural nets and HMMs. This set-up allows to monitor the change in recognition accuracy, if the feature set is changed, but the HMM set is kept fix and vice versa. The HMMs are trained using non-quantized features, thus the training uses a non-distributed environment. The three test sets are also taken from the AURORA database. Each test set contains speech added with noise from known and unknown sources at different SNRs. Test A contains the same noise types that appear in the multi-condition training set, test B contains unknown noise types and test set C contains known and unknown noise filtered with the MIRS channel characteristic (the training set is filtered with the G.712 characteristic, for more details about the filter characteristics see [2]).
Table 1 shows the results of the 3 feature sets with the corresponding neural nets and the HMM set trained on the system with MFCC features only. Table 2 includes the same 3 feature sets (again with the corresponding neural nets) and the HMM set trained on the system with RASTA-PLP features only. Finally, table 3 uses a HMM set trained with the combined feature vector MFCC+RASTA-PLP. The figures denote the relative loss or improvement compared to the reference recognition results obtained with a standard HMM system in [2]. The neural networks use the actual frame plus 6 context frames, all features are computed with delta and acceleration coefficients. Comparing the results we can state the following observations:

- the best result averaged over all three test sets can be achieved with NN and HMM trained with RASTA-PLP features
- the NN trained with RASTA-PLP features is best suitable for unknown noise conditions or acoustic channel distortions, the combination RASTA-PLP+MFCC is best suitable for known noise conditions
- the choice of features used in the HMMs' training step has no significant impact on the recognition result

- apart from the pure MFCC system (that only outperforms the standard approach under known noise conditions), an overall gain (through all test sets) compared to a standard HMM recognizer is noticeable

Summarizing these observations we can stress the suitability of the hybrid framework for DSR. Once a system is built up, the client and server components can be tuned to fulfill the task's requirements without rebuilding the whole system. The feature extraction method can be adapted to the expected noise conditions. Using the investigated feature types we would use a NN with RASTA-PLP features if the microphone or the environment changes and another one with MFCC+RASTA-PLP features if the noise and channel conditions stay the same[3]. The server side is untouched during this adaptation, this would not be possible in a standard distributed system. As long as the set of posteriors is kept fix we can also change the server side regarding HMM topology this would not be possible in a hybrid system based on the approach presented in [4].

NN input layer	Test A	Test B	Test C
MFCC	17.45%	-19.35%	-30.25%
RASTA-PLP	14.1%	4.52%	29.24%
MFCC+RASTA-PLP	32.29%	0.59%	2.55%

Table 1. HMM trained with MFCC features, relative deviation to the baseline recognition result (Gaussian HMM recognizer)

NN input layer	Test A	Test B	Test C
MFCC	16.43%	-23.61%	-31.23%
RASTA-PLP	15.08%	5.44%	30.52%
MFCC+RASTA-PLP	31.88%	-0.4%	1.36%

Table 2. HMM trained with RASTA-PLP features, relative deviation to the baseline recognition result (Gaussian HMM recognizer)

NN input layer	Test A	Test B	Test C
MFCC	16.63%	-26.11%	-34.36%
RASTA-PLP	14.97%	3.83%	29.4%
MFCC+RASTA-PLP	32.09%	-1.34%	1.52%

Table 3. HMM trained with MFCC+RASTA-PLP features, relative deviation to the baseline recognition result (Gaussian HMM recognizer)

[3] comparing training and test conditions

5. CONCLUSION

We have explored a distributed hybrid speech recognition system using different feature extraction algorithms. In particular, MFCC features, RASTA-PLP features and the combination MFCC+RASTA-PLP have been investigated. We outlined the advantages of the hybrid approach in a distributed environment concerning the generalization of the system design. Our experiments using the AURORA 2 database show that training a NN with RASTA-PLP features on the client side outperforms a standard system based on Gaussian densities under all test conditions. If the noise conditions are known in advance, the combination MFCC+RASTA-PLP produces even better results. The hybrid DSR system can be adapted in a flexible way to new environments: New features can be added, others removed and the amount of context is freely adjustable on the client side since the interface to the channel does not notice any of these changes and the server side is kept unchanged.

6. REFERENCES

[1] 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.

[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] 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.

[4] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[5] J. Rottland and G. Rigoll, "Tied posteriors: An approach for effective introduction of context dependency in hybrid NN/HMM LVCSR," in *Proc. ICASSP*, 2000.

[6] H. Hermansky, "Perceptual linear predicitive (PLP) analysis of speech," *Journal of the Accoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

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