Vysoké učení technické v Brně
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií

Ing. Petr Schwarz

Rozpoznávání fonémů z dlouhého časového okolí

Phoneme recognition based on long temporal context

Zkrácená verze disertační práce

Obor: Informační technologie

Školitel: Doc. Dr. Ing. Jan Černocký

Klíčová slova: rozpoznávání fonémů, TIMIT, neuronové sítě, časové trajektorie, dlouhý časový kontext, dělený časový kontext, identifikace jazyků.

Keywords: phoneme recognition, TIMIT, neural networks, temporal patterns, long temporal context, split temporal context, language identification.

Rukopis disertační práce je uložen na Fakultě informačních technologií Vysokého učení technického v Brně, Božetěchova 2, 61266 Brno. Plný text disertační práce je k disposici na:
`http://www.fit.vutbr.cz/˜schwarzp/publi/thesis.pdf`

Contents

Chapter 1

Introduction

Phoneme recognition is very important part of automatic speech processing. Phoneme strings can transcribe words or sentences and the storage space is very small. It can be applied in many areas of speech processing – in large vocabulary continuous speech recognition, keyword spotting, language identification, speaker identification, topic detection, or in much easier tasks like voice activity detection. N-grams of phonemes are easily indexable, therefore phoneme recognition can be a basic part of systems for search in voice archives. In phonotatic language identification, topic detection or speaker identification, the language, topic or speaker can be represented by a phonotactic ”language” model modelling dependencies among phonemes in phoneme strings. The accuracy of phoneme recognizer is crucial for the accuracies of all the mentioned technology! Therefore it is worth to investigate phoneme recognition and it is worth to develop as accurate phoneme recognizer as possible.

The thesis is focused on the main part of phoneme recognition, on acoustic modelling techniques. There are many other related issues, like channel normalization, channel and speaker adaptation, multilinguality, robustness in noise, but these issues are not investigated in detail

People recognize words from quite long temporal context. Sometimes we realize what was said even after few seconds, minutes or days. It depends on the quality and complexity of a model of the world we have in our heads. We are still far away form such model. This work investigates a basic model of phoneme and it tries to get as much as possible from the contextual information. Much longer temporal context than usual is used.

The main effort is given to a hybrid Artificial Neural Network / Hidden Markov Model approach.

1.1 Motivation

The main motivation for this work is the wide range of applications/tasks that the phoneme recognition affects. Improving phoneme recognition is not linked to just one particular problem but to wide ranges of problems. Phoneme recognition is not a closed box. It can be seen as an application of investigated acoustic modelling techniques. A better understanding of these techniques can allow us to better react to other needs in speech processing.

Another motivation was my study and then employment in Speech@FIT speech processing group at Brno University of technology and a stay at Oregon Graduate Institute. The groups were already investigating speech modelling techniques and features based on a long temporal context. But that time the techniques were used almost blindly. Deeper understanding helped to speed up the research and motivated research in another areas.

1.2 Original claims

In my opinion, the original contributions – “claims of this thesis” can be summarized as follows:

- Extensive comparison of phoneme recognition systems based on different structures of Artificial Neural Networks (ANN) and Gaussian Mixture Models (GMM).
- Detailed study of Temporal Pattern (TRAP) based system and its simplification.
- Definition of a split temporal contexts (STC) system reaching very good phoneme recognition results.
- Tuning of phoneme recognizers – applying and studying common speech recognition techniques that can decrease the phoneme error rate.
- Studying of phoneme recognizers on different databases, with varying amounts of training data, in noise and in cross-channel condition.
- Application of the long temporal context based phoneme recognizer to language identification, keyword spotting and voice activity detection.
- Discussion about techniques that can help to accurately train neural networks in speech recognition.

Chapter 2

Baseline systems

This chapter concentrates on the basic phoneme recognition experiments with HMM/GMM and HMM/ANN systems and novel TRAP based techniques. In order to be comparable to state-of-the-art, all results are reported on TIMIT.

2.1 What system as baseline?

The Temporal Pattern (TRAP) system was taken as baseline. This system was known to give better results than conventional techniques (HMM/GMM with MFCC) in some cases [7] (mainly in cross-channel conditions), and the ANN features (posterior probabilities of phonemes) were known to be complementary features for MFCCs or PLPs [9]. But there was no detailed understanding of the whole approach, therefore the TRAP system is studied. The TRAP system is compared to some conventional systems based on MFCCs. There is a big step between the TRAP system based on HMM/ANN hybrid and a HMM/GMM based on MFCCs, therefore the HMM/ANN and HMM/GMM are compared on MFCC features at first and then the TRAP system is compared to a HMM/ANN hybrid based on MFCCs.

2.1.1 HMM/GMM

All the GMM experiments are done with the HTK toolkit[1]. The features are $MFCC+C_0+\Delta+\Delta\Delta$ (together 39 coefficients). This feature set is referred as MFCC39. The HMM models were initialized to global means and variances. Then the models were re-estimated, all the Gaussians split to two and re-estimated again. This was repeated up to 256 Gaussians. The recognition was done using the HVite decoder.

2.1.2 HMM/ANN

The HMM/ANN hybrid is based on the SVite decoder and the QuickNet ANN software[2]. The SVite decoder is a part of BUT STK toolkit[3]. The input features are $MFCC + C_0 + \Delta + \Delta\Delta$ or other features derived from Mel-bank energies in later experiments. Neural networks are trained to map input features to phoneme posteriors according to hard labels (each feature vector is assigned to one phoneme).

[1] `http://htk.eng.cam.ac.uk`

[2] `http://www.icsi.berkeley.edu/Speech/qn.html`

[3] `http://speech.fit.vutbr.cz/en/software/hmm-toolkit-stk-speech-fit`

2.1.3 HMM/GMM and HMM/ANN based on MFCCs with one state model

This experiment compares HMM/GMM system and HMM/ANN hybrid. The input features are MFCC39. The numbers of parameters in GMM or ANN were found such way that the decrease in phoneme error rate caused by adding new parameters is negligible. This procedure was used also in all following experiments. The final number of Gaussian mixtures is 256 and final number of neurons in the hidden layer is 500. The results are in Table 2.1. There is almost no difference in PER (0.3 %), so if the features are well adapted to the model and the training procedure is optimal, it should be possible to reach similar results with both HMM/GMM and HMM/ANN systems. Table 2.2 shows the number of parameters in both systems. The HMM/ANN system 5 % parameters compared to the HMM/GMM system.

system	ins	sub	del	PER
GMM	4.1	18.7	15.2	38.0
AMM	4.7	20.6	12.4	37.7

Table 2.1: *Comparison of HMM/GMM and HMM/ANN based on MFCCs with one-state model.*

system	# parameters (floating point numbers)
GMM	788736
NN	39539

Table 2.2: *Comparison of numbers of parameters in HMM/GMM and HMM/ANN systems based on MFCCs.*

2.2 Basic TRAP system

The TRAP system is shown in detail in Figure 2.1. Speech is segmented into frames 25 ms long and for each frame, mel-bank energies are calculated. Temporal evolution of energy for each band is taken (101 values = 1 second), normalized to zero mean and unit variance across the temporal vector, windowed by Hamming window and then normalized to zero means and unit variances across all training vectors. This is beneficial for the ANN as it is ensured that all inputs have the same dynamics. For testing, the later normalization coefficients are not calculated but taken from the training set. Such prepared temporal vectors are presented to band neural networks. These neural networks are trained to map temporal vectors to phonemes. A vector of phoneme posterior probabilities is obtained at the output of each band neural network. The posterior probabilities from all bands are concatenated together, the logarithm is taken and this vector is presented to another neural network (merger). The merger is trained to map the vectors to phonemes again. The output is a vector of phoneme posterior probabilities. Such vectors are then sent to the Viterbi decoder to generate phoneme strings.

2.2.1 Effect of mean and variance normalization of temporal vector

The mean and variance normalization of temporal vector makes the TRAP system more robust against channel change. The normalizations works similarly as cepstral mean and variance normalization[4], commonly applied in MFCC.

[4] Cepstral coefficients are extracted from Mel-bank energies by the DCT transform. DCT is a linear transform.

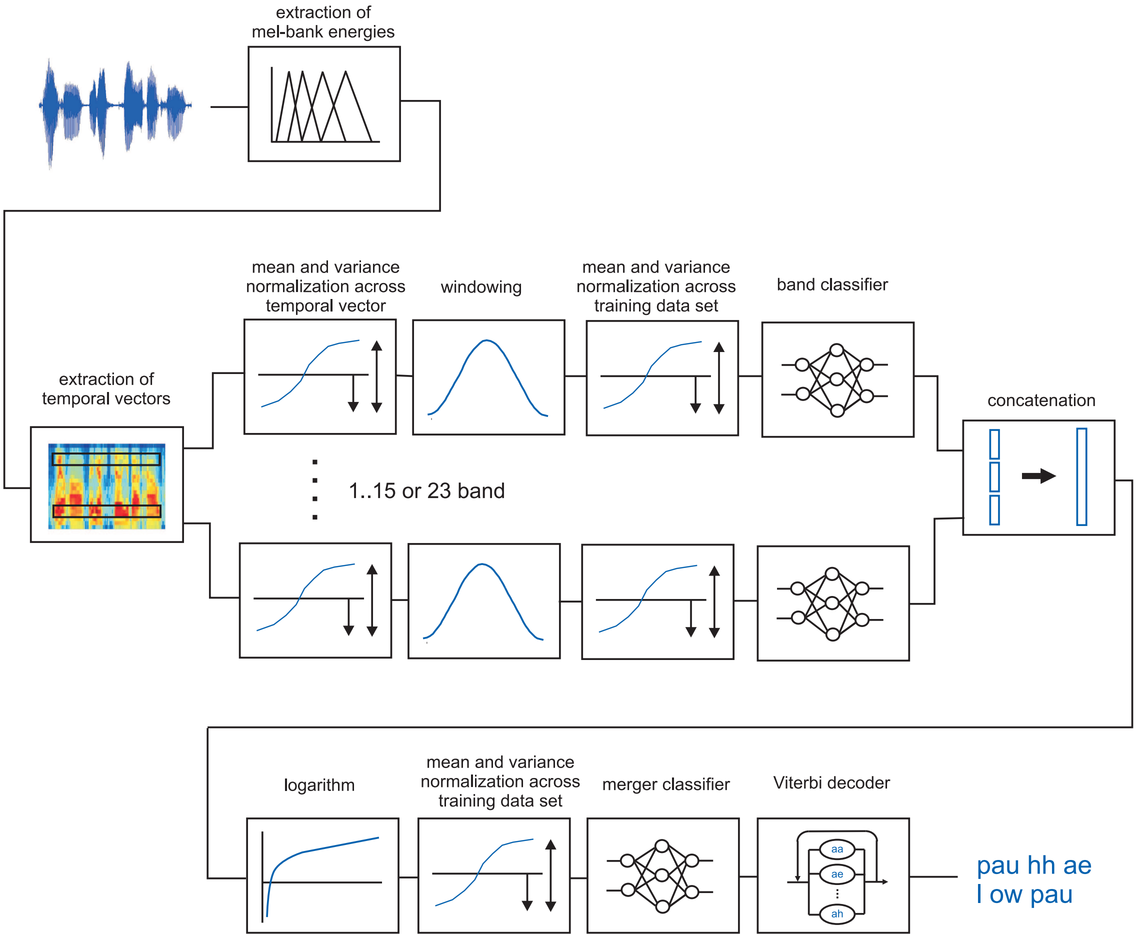

Figure 2.1: *Block diagram of the TRAP system.*

The mean normalization can be seen also as a temporal filtering, similar to RASTA [5]. A change in the length of temporal vector affects characteristics of the filter. There is no visible benefit from tying the temporal vector length and the window length for mean and variance normalization. Both can be tuned separately. This normalization was disabled in performed experiments. The main focus of this thesis is on the acoustic modelling and if this normalization is applied, it can influence other parameters, mainly the optimal length of temporal context.

2.2.2 Windowing and normalization across the data set

The window used to select the trajectory out of the evolution of critical band energy has no effect in the TRAP system. The window is canceled out by the mean and variance normalization across the training data set:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{2.1}$$

where $\tilde{\mathbf{x}}$ is normalized vector, $\mathbf{x}$ is input vector. $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are mean vector and vector of standard deviations, both estimated from all vectors in the training set:

$$\boldsymbol{\mu} = \frac{1}{F} \sum_{i=1}^{F} \mathbf{x}_i \tag{2.2}$$

$$\boldsymbol{\sigma}^2 = \frac{1}{F}\sum_{i=1}^{F}(\mathbf{x}_i - \boldsymbol{\mu})^2 = \frac{1}{F}\sum_{i=1}^{F}\mathbf{x}_i^2 - \left(\frac{1}{F}\sum_{i=1}^{F}\mathbf{x}_i\right)^2 \tag{2.3}$$

F is the number of frames in the training set. When equations 2.2 and 2.3 are substituted into equation 2.1, it can be easily seen that both vectors with weighting window applied and without weighting window are equal after normalization.

$$\frac{\mathbf{wx} - \frac{1}{F}\sum_{i=1}^{F}\mathbf{wx}_i}{\sqrt{\frac{1}{F}\sum_{i=1}^{F}\mathbf{w}^2\mathbf{x}_i^2 - \left(\frac{1}{F}\sum_{i=1}^{F}\mathbf{wx}_i\right)^2}} = \frac{\mathbf{x} - \frac{1}{F}\sum_{i=1}^{F}\mathbf{x}_i}{\sqrt{\frac{1}{F}\sum_{i=1}^{F}\mathbf{x}_i^2 - \left(\frac{1}{F}\sum_{i=1}^{F}\mathbf{x}_i\right)^2}} \tag{2.4}$$

w is weighting window vector.

2.2.3 Mean and variance normalization across the data set and ANN training

This normalization is related to the training of neural networks. The main benefit from it is a faster training and reduced chance to get stuck in a local minima during criterial function optimization. The effect of the normalization is illustrated in Figure 2.2. Let's suppose just one neuron with two inputs:

$$y = \mathcal{F}(w_1x_1 + w_2x_2 + t) \tag{2.5}$$

where y is the output of the neuron, $\mathcal{F}$ is a nonlinear output function (for example sigmoid), w_1 and w_2 are weights (we want to train them), x_1 and x_2 are inputs and t is a threshold (also trained). The weights and the threshold are set randomly in certain dynamic range at the beginning of the training. The dashed line defining division of space[5] by the neuron is:

$$\begin{array}{c} 0 = w_1x_1 + w_2x_2 + t \\ x_1 = -\frac{w_2}{w_1}x_2 - \frac{t}{w_1} \end{array} \tag{2.6}$$

The threshold t moves the discrimination line up and down. The points represent two classes in feature space (for example phonemes). They can be far away from the center of axes if no normalization is applied. It is necessary to run many training iterations to move the discrimination line closer to the data clusters (Figure 2.2a). If only mean normalization is applied, the dynamic range of weights and threshold is not necessarily appropriate to the dynamic range of data. The discrimination line can be even out of the data points and again, the training will need more iterations (Figure 2.2b). Figure 2.2c shows the data points after mean and variance normalization.

2.2.4 Comparison to systems based on classical features

In this section the TRAP system is compared to two classical systems: a hybrid system based on MFCC39 and a hybrid system based on multiple frames of MFCC39. The optimal number of consequent frames for the multiframe system was experimentally found to be 4. The TRAP system reached slightly better PER than the MFCC39 multiframe system, but the improvement to the pure MFCC39 system is significant. The results are in Table 2.3.

[5] In this case a 2D plane

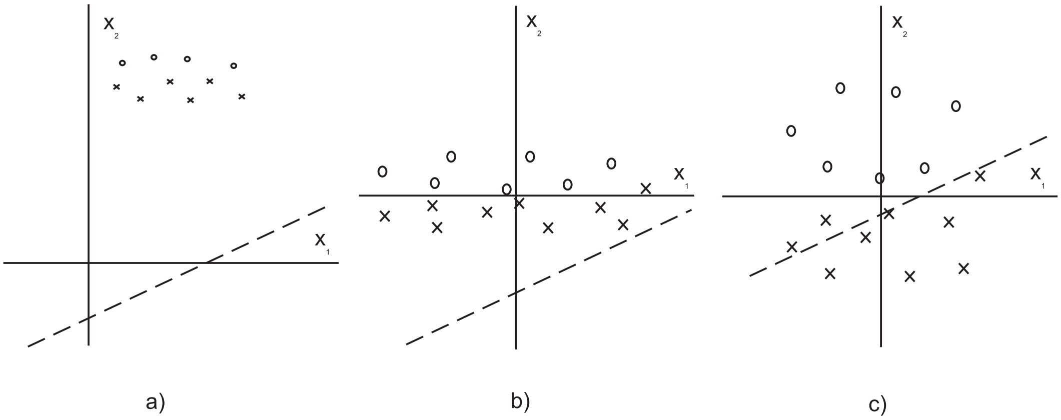

Figure 2.2: *Effect of mean and variance normalization across data set – two class example: a) without normalization, b) with mean normalization, c) with mean and variance normalization.*

	ins	sub	del	PER
MFCC39	4.7	20.6	12.4	37.7
MFCC39 – 4 frames	5.5	19.0	9.6	34.1
TRAPS – 1 second	4.3	18.6	10.9	**33.8**

Table 2.3: *Comparison of systems based on MFCC to the TRAP system.*

2.2.5 Optimal length of temporal context

The one second long temporal context, usually used in literature [6], is not necessarily optimal.

Some weights of neural networks could be uselessly spent on parts of temporal context with a little relevant information. We may also not have enough training data to extract this information. Therefore the optimal length was found experimentally. The length of TRAP is being increased from 100 ms to 1 second and the PER is evaluated.

It is very important not to use mean and variance normalization of temporal vector for this experiment. These normalizations dramatically increase PER for short contexts and bias the experiment.

Table 2.4 shows the results. The optimal length is about 300 ms ÷ 400 ms. It means using 150 ms ÷ 200 ms to the future and 150 ms ÷ 200 ms to the past. The optimal temporal context length is shorter than the 1 s used by other authors. The fact that shorter input is effective may have positive implications in applications where minimal algorithmic delay is required. During other experiments not described here, the optimal length was found to depends on task (it is longer for digit recognition), on the size of neural network and on the amount of the training data. The PER is already much better than for the MFCC39 multiframe system which is a proof that longer temporal context is usefull.

length (ms)	110	210	310	410	510	610	710	810	1010
PER (%)	33.6	**31.3**	**31.3**	**31.3**	31.6	32.0	32.2	32.6	33.8

Table 2.4: *Effect of temporal context length in the TRAP system.*

2.2.6 Discussion

The main motivation for the TRAP system presented in literature is greater robustness against channel change and noise due to independent processing of frequency band and ability to extract information from a longer temporal context.

The later motivation was verified to be correct. The longer temporal context brings new information and moves data points representing different phonemes further apart in feature space. Therefore the system is more robust.

The former motivation was not verified yet. The greater robustness can come from mean and variance normalization of temporal vectors (not applied here). But the normalization can be done separately on in the structure of a classifier. The hierarchical structure of neural networks can still perform just a nonlinear mapping function. It is not able to find which information is incorrect and selectively discard this information.

The purpose of band neural networks needs a deeper investigation. The experiments indicate that the purpose of these nets is not classification to phonemes for a simple decision in merger, but rather a data preprocessing for merger. Otherwise the optimal temporal context length would be similar for both the bands and the whole system.

2.3 Simplified system (one net system)

The TRAP system is complex and runs slowly. Even the experiments are slow, therefore the TRAP system is simplified. The simplification is necessary also for a better understanding of the whole system.

The band neural network represents a nonlinear mapping function. Let us replace this nonlinear function with a linear one: a linear transform is estimated instead of neural network weights and biases. And let's go further and omit the mapping to phonemes. The assumption is that the useful information is characterized by a variance in data. The Principal Component Analysis (PCA) is used to estimate the linear transform. One transform is estimated for each band. The base components are very similar to Hamming window weighted Discrete Cosine Transform (DCT) bases, therefore a simplification to DCT was also tested. An experiment confirmed that the DCT degraded the results negligibly, therefore the DCT transform is used in the following experiments. A dimensionality reduction follows the linear transform. Network training can be helped by optimal choice of the dimensionality of input feature vector.

	ins	sub	del	PER
simplified system	3.7	16.6	9.6	**29.9**
TRAP system	4.1	17.4	9.8	31.3
TRAP + DCT	4.0	17.3	9.8	31.1

Table 2.5: *Comparison of simplified system and the TRAP system.*

Comparison of the simplified system to the TRAP system in terms of PER can be seen in Table 2.5. The length of temporal vectors is 310 ms and 16 DCT coefficients were kept. The experiment showed that the linear transformation is enough. The simplified system gives even better results than the complex TRAP system.

For investigation of the effect of nonlinear transforms in bands, the DCT and dimensionality reduction were applied also before band neural networks in the TRAP system. This was done previously by František Grézl but without any explanation [3][2]. This approach reached better result than the TRAP system but worse than the simplified system. This could mean that band neural networks do something similar as chain of windowing, DCT and dimensionality reduction. This chain is discussed thoroughly in the following subsections.

2.3.1 Weighting of temporal vectors and DCT

The weighting of temporal vectors has no effect in the TRAP system. It was canceled out by the subsequent normalization. The situation changed in the simplified system, the weighting start to be beneficial. Let us see an experiment. The simplified system was trained with and without DCT and with or without Hamming window. The results are in Table 2.6.

The first two rows indicate that it does not matter whether the window is applied or not if the DCT is not applied. Precisely, the result with window is even worse, but this can be just a bad luck as the training algorithm got stuck in a local optimum. If the DCT is applied (third row), the result is significantly better. The improvement comes from smaller patterns (less parameters at the input of the network). The dimensionality reduction implies the fact that the temporal trajectory can be down-sampled twice without any degradation in accuracy. This had been already found in [8] and [4]. The DCT with dimensionality reduction can be also seen as a kind of temporal filtering, similar to RASTA [5]. Here, smaller and smoother patterns imply less trainable parameters in the neural network and less chance to get stuck in local optimum during the training. If the window is applied together with DCT (last row), the result is even better. The DCT saved the window and it was not canceled out by the normalization! The window attenuates values at the edges of temporal context, so the training algorithm can focus to the center of the context during the initial phase of training.

	ins	sub	del	PER
no window, no DCT	4.2	18.0	10.4	32.6
Hamming, no DCT	4.0	18.5	10.5	33.0
no window, DCT	4.2	17.3	9.2	30.7
Hamming and DCT	3.7	16.6	9.6	**29.9**

Table 2.6: *Effect of windowing of temporal vectors (PER).*

Why is the attenuation important? The answer can be found in histograms of values at different places of the temporal vector. The histogram is narrow for the center (the variance is low). Then the width grows and it is the highest at the edges. The trajectory in feature space representing a phoneme is affected by neighboring phonemes. The DCT tries to describe the input pattern by first few bases in such a way that the variance in the pattern is preserved. The DCT features must be definitely focused to the edges if no window is applied. The window allows to describe the central part of context with a better resolution.

2.3.2 The Discrete Cosine Transform as a frequency filter

The DCT applied to temporal vector can be seen as a modulation frequency band-pass filter. What are the important frequencies that needed to be modelled? The lower frequency limit is given by the length of temporal vector. If the length is higher, lower frequencies can be modelled. The upper frequency limit is given by the number of used DCT coefficients. But the number required DCT coefficients to keep a constant upper frequency limit grows also with the length of the temporal vector. Is it better to keep the input for neural network constant and model narrower frequency range for longer context, or is it better to increase the input and keep the frequency range constant? The following experiment gives answers to these questions. The optimal length of temporal context is evaluated for fixed number of DCT coefficients (15 + C_0) and then the number of DCT coefficients is varied according to equation:

$$n_{DCT} = \frac{context\_length}{2} + 1 \tag{2.7}$$

This equation ensures fixed upper frequency limit. The context length is in frames (10 ms units). The results are in Table 2.7 and in Figure 2.3. Both lower frequency and upper frequency limits are reported. The optimal length of temporal contexts is about 300 ms for both cases. This is similar as for the TRAP system. It is definitely better to keep the upper frequency limit constant (to increase the number of DCT coefficients), as can be seen from the figure. It is possible to get an additional information using a longer temporal context, but it is necessary to model the whole trajectory with equal variance (detail) as before.

	length (ms)	110	210	310	410	510	610	710	810	1010
	lower f_m (Hz)	4.6	2.4	1.6	1.2	1.0	0.8	0.7	0.6	0.5
fixed	upper f_m (Hz)	68.2	35.7	24.2	18.3	14.7	12.3	10.6	9.3	7.4
# DCT	# DCT	16	16	16	16	16	16	16	16	16
	PER (%)	-	-	**29.9**	30.2	30.8	32.4	33.9	35.7	39.6
varied	upper f_m (Hz)	22.7	23.8	24.2	24.4	24.5	24.6	24.7	24.7	24.8
# DCT	# DCT	6	11	16	21	16	31	36	41	51
	PER (%)	34.5	30.8	**29.9**	30.4	30.6	30.8	31.5	31.3	32.4

Table 2.7: *Effect of temporal context length for fixed and varying number of DCT coefficients (the f_m is modulation frequency).*

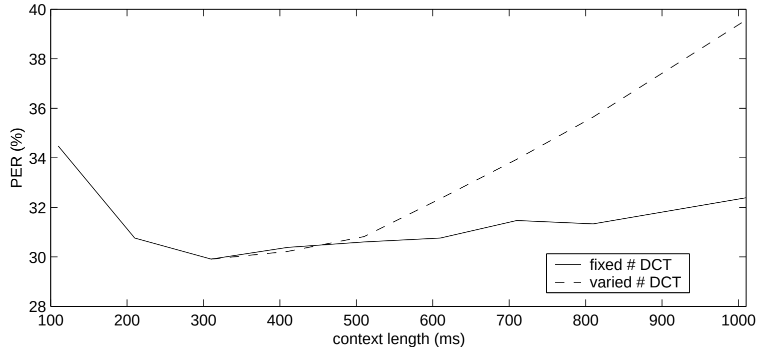

Figure 2.3: *Dependency of PER on temporal context length for fixed and varied number of DCT coefficients.*

2.4 Study of amount of training data

The phonemes are represented by trajectories in the feature space. There is not one trajectory for one phoneme, but many. The number grows with the length of temporal context. Let us consider one phoneme: this phoneme can be affected by 39 phonemes on the left and by 39 phonemes on the right. Each of these phonemes can be affected by 39 others. The number of trajectories will grow exponentially.

Let's study the amount of data we have in the database for certain lengths of the temporal context. The average phoneme length is a good unit for measurement. The task can be simplified and the n-grams statistics can be used[6].

Table 2.8 shows the coverage of n-grams in the test part of TIMIT database. The most important columns are the third (numbers in brackets) – percentage of n-grams occurring in the test part but not in the training part, and fourth – error which would be caused by a decoder if the unseen n-grams

[6] Note that we never use those n-grams in phoneme recognition, it is just a tool to show amounts of sequences of different lengths!

are not allowed. The error is calculated by sum of occurrences of unseen n-grams divided by sum of occurrences of all n-grams:

$$error = \frac{\sum_{i \in N} C\left(n_i\right)}{\sum_{i \in A} C\left(n_i\right)} \tag{2.8}$$

N is a set of unseen n-grams, A is a set of all n-grams and $C\left(n_i\right)$ gives number of occurrences of n-gram n_i in the test part of database.

For bi-grams, there are 2.26 $\%$ of unseen cases but this amount causes almost no error (0.13 $\%$). The situation is much worse for trigrams with 18.83 of unseen cases causing 7.60 $\%$ of error. It is almost impossible to model four-grams due to 44.10 $\%$ of error. These errors can be expected to be smaller in case of larger databases but still the maximum possible length of context seems practically to be three times or four times phoneme length due to exponential growth of error.

To conclude, the most limiting issue for a system based on long temporal context is the amount of training data because the demand for data grows exponentially with the temporal context length. This situation force us to look for a way around. One solution is to collect huge databases. Current systems use more than 1000 hours of training data [1]. This system just 2.5 hours. The collecting and annotation of new databases is very costly. But it is the mostly used way today. Another solution is the development of clever algorithms. This way is chosen for this thesis.

n-gram order	# different n-grams	# not seen in the train part	error (%)
1	39	0 (0.00%)	0.00
2	1104	25 (2.26%)	0.13
3	8952	1686 (18.83%)	7.60
4	20681	11282 (54.55%)	44.10

Table 2.8: *Numbers of occurrences of different n-grams in the test part of the TIMIT database, number of different N-grams which were not seen in the training part and error that would be caused by omitting unseen N-grams in the decoder.*

Chapter 3

System with split temporal context (LC-RC system)

3.1 Motivation

The study of amount of data needed to train a long temporal context based system (section 2.4) showed that very large databases are necessary. A development of techniques that need less data and limit the cost spent on data collection and annotation would be beneficial. This chapter investigates one such technique. This technique is inspired by the function of band neural networks in the TRAP system and Table 2.8.

If we are not able to classify long trajectories in the feature space because there are simply many of them and very big portion was not seen during training, let us to split the trajectores into more parts.

These parts can be modelled separately and then the results can be merged together. An assumption of independence is done. Obviously by the split, a part of information is lost.

Let us see what will happen if the trajectory is split into two parts on n-gram statistics. All trigrams were split into two bigrams. The error caused by unseen trigrams 7.60 % was replaced by two times the error of bigrams which is only 2 × 0.13 % = 0.26 %. For four-grams, the error was reduced from 44.10 % to just 15.2 %. The reduced errors are summarized in Table 3.1.

n-gram order	# different n-grams	# not seen in the train part	error (%)	reducted error (%)
2	1104	25 (2.26%)	0.13	0.00
3	8952	1686 (18.83%)	7.60	0.26
4	20681	11282 (54.55%)	44.10	15.2

Table 3.1: *Effect of splitting trajectories into two parts – reduced errors. All other columns are unchanged.*

3.2 The system

The experimental system is derived from the simplified system described in section 2.3. The Mel-bank energies were extracted and the 310 ms long temporal vectors (31 values) of evolution of critical bank energies were taken. Each temporal vector was split into two parts – left part (values 0 - 16) and right part (values 16 - 31). Both parts were windowed by corresponding half of Hamming window and projected to the DCT bases. 11 DCT coefficients were kept for each part. Such preprocessed vectors were concatenated together for each part of context separately and sent to two neural networks – these are trained to produce phoneme posteriors, similary as in the TRAP system. Output posterior vectors are concatenated, transformed by logarithm and sent to another (merging) neural network trained again to deliver phoneme posteriors. Finally, the phoneme posteriors are decoded by a Viterbi decoder and strings of phonemes are produced. The whole process is illustrated in Figure 3.1. This system is called the Left context – Right context system, or shortly LC-RC system.

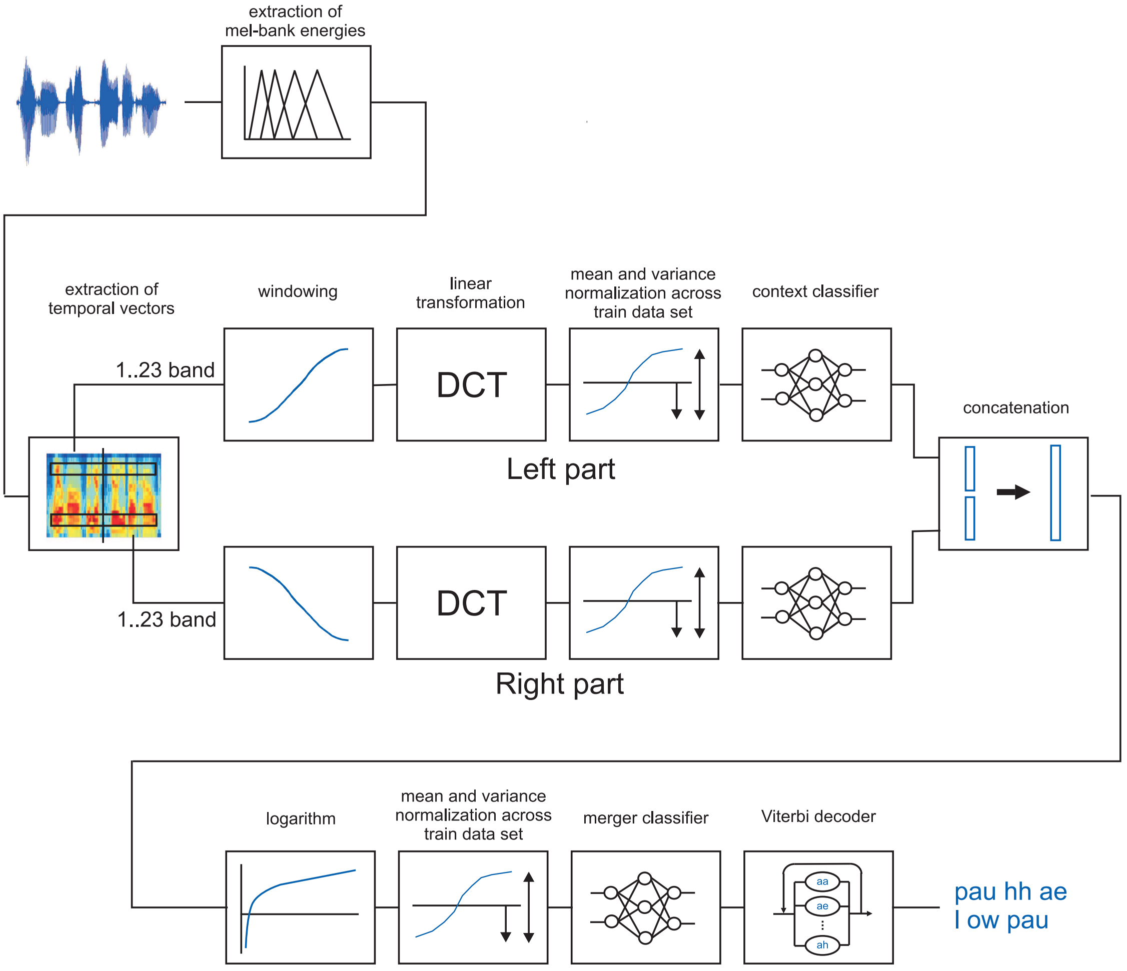

Figure 3.1: *Block diagram of the Split Temporal Context system.*

3.3 First result and comparison to the simplified system

The LC-RC system was compared to the simplified system. The results are in Table 3.2. The RC-LC system reached significanlty better result. The motivation was proven to be correct despite the independence assumption.

system	ins	sub	del	PER
simplified	3.7	16.6	9.6	29.9
LC-RC	4.0	15.4	9.0	**28.4**

Table 3.2: *Comparison of the LC-RC and simplified systems.*

3.4 Modelled modulation frequencies

Where the improvement in the LC-RC systems comes from? The following experiment tries to answer the question. The optimal number of DCT coefficients was found for the left context. Table 3.3 shows the results. It is the best to include 14 DCT coefficients (almost all).

Now let us compare modulation frequencies modelled by both the LC-RC and the simplified systems. The comparison is in Table 3.4. The upper limit for modelled frequencies is much higher for the LC-RC system. The LC-RC system models the trajectory with lower variance (higher details). The two blocks also increase the temporal resolution. The remaining question is the drawback of the LC-RC system coming from not seeing frequencies bellow 1.67 Hz. Removing C_0 in the simplified system causes increment in PER as the information about vertical shifts in different bands is lost and is not seen by the network. However the lost in PER does not exeed 0.5 %.

# coef upper f_m (Hz)	6 16.7	8 23.3	9 26.7	10 30.0	11 33.3	12 36.7	13 40.0	14 43.3	16 50.0
PER (%)	36.9	36.0	35.7	35.7	35.4	35.5	35.4	**35.2**	35.4

Table 3.3: *Optimal number of DCT coefficients (including C_0) for the left context in the LC-RC system and corresponding modulation frequencies.*

system	context length (ms)	optimum # coefs (-)	lower f_m (Hz)	upper f_m (Hz)
simplified system	310	16	1.67	25.00
LC part	160	14	3.33	43.33

Table 3.4: *Comparison of minimal and maximal modulation frequencies for the left part in the LC-RC system and the simplified system.*

3.5 Optimal lengths of left and right contexts

The previous experiment showed that the upper limit of modulation frequency used by the LC-RC is significantly higher than for the simplified system. If we have a more capable classifier, is not it worth to extend also the temporal context? At first, let us evaluate the optimal temporal context length for context networks. The results are in Table 3.5. The number of DCT coefficients was set according to equation:

$$n = \mathrm{int}\left(\frac{2}{3}\frac{len}{10}\right) \tag{3.1}$$

This equation ensures the upper limit of modulation frequencies constant (about 33 Hz). The operator ”int” is rounding to the first lower integer.

len (ms)	100	120	140	160	180	200	220	240	260	310	360	410
LC PER (%)	36.5	36.0	36.3	35.4	35.8	**35.1**	35.2	35.4	35.8	35.9	35.8	36.5
RC PER (%)	37.8	38.0	37.4	37.7	37.2	**37.1**	37.2	37.5	37.3	37.4	37.7	38.1

Table 3.5: *Optimal length of left and right temporal contexts in the LC-RC system.*

The minima for both contexts are at 200 ms. This is interesting, because the full context is about 400 ms which is closer to the optimum for band neural networks in the TRAP system seen in section

len (ms)	270	310	350	390	430	470	510	550
13 DCTs	28.5	27.9	27.8	27.8	27.8	**27.6**	27.8	27.9
16 DCTs	-	28.3	28.0	27.8	27.9	**27.6**	27.8	27.8

Table 3.6: *Optimal length of temporal context for the whole LC-RC system.*

??, where we know that useful information for classification is contained. The beginning of both graphs in Figure **??** seems to be quite noisy. The peaks partially disappear if more DCT coefficients are used. This suggests that the DCT transform is not the best choice to model higher modulation frequencies. The PER is better for the left contexts. This indicates that the signal at the beginning of phoneme is more important.

3.6 Optimal length of temporal context for the whole LC-RC system

An optimistic result from the previous section does not ensure that the whole system will use all the 400 ms given by sum of both optimal context lengths. Therefore the same experiment was repeated for the whole system. Both contexts have the same length. This time, the number of DCT coefficients was fixed to ensure stability in the initial part of graph. The results are in Table 3.6. The optimal length is even higher than the sum of optimal lengths for both contexts! The optimal lengths of contexts for merging differ from the context lengths with minimal PER. The final part of the graph (crossing lines) shows again that it is important not to cut off the upper modulation frequencies.

3.7 Discussion

This chapter proved that the information usable for recognition of a phoneme is spanned across almost 500 ms. And we are actually able to extract the information! This chapter also brought more insight to the training of neural networks. It is beneficial to introduce some reasonable constrains coming from the task.

Although we see the optimal parameters, the later experiments are done with a shorter temporal context (310 ms) and less number of DCT coefficients (11 per context). The reason is comparison with the baseline systems, and also a faster turnover of experiments.

Chapter 4

Towards the best phoneme recognizer

The previous two chapters described the development of a good phoneme recognizer. The next goal described in this chapter was is to improve it as much as possible by adding techniques commonly used in speech recognition.

4.1 More states

One of the most common techniques in speech recognition are state models. The main purpose of states is the selection of particular views at features. The decoder proposes new direction of trajectory and a model in a state verifies whether the direction is correct or not. It is similar as if someone advices us a route. We can verify that we are still on the route according to some important objects at different places of the route. The more important objects we see the more sure we are.

Features based on long temporal context and ANN can see many important objects from one point. But the main benefit of states comes during the training. The training algorithm is focused to certain parts of phonemes. We guide the training. The focused patterns are easier and sharp. The weights are associated with certain parts of phoneme and in case of increasing the number of parameters of the network, we have a chance to decrease the error.

The hierarchical structure of neural networks also benefits from states during recognition. The lower network (band or context network) roughly estimates a place (state) where the recognition is in the feature space. The position is more precise with more states. The upper network (merger) uses the knowledge about this place and it can focus on details.

Another benefit is minimum phoneme duration. If three state models are used, the minimum duration of phoneme is 30 ms. This is good to prevent the decoder from switching of one phone to another, although this can be also enforced by repetition of existing HMM states or setting appropriate phoneme insertion penalty.

4.1.1 Implementation of states

The parametrization and neural network structure is unchanged for this approach. The neural networks were trained on force-aligned state transcriptions. The decoder was modified to force a pass through the state sequences in phoneme models. The phoneme models are left-to-right with no skip states.

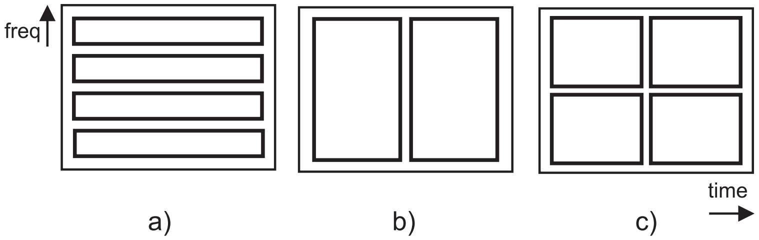

Figure 4.1: *Different time and/or frequency split architectures: a) TRAP system, b) LC-RC system, c) 2 x 2 system*

4.1.2 Results

The realignment does not bring any improvement for one-state models. Three iterations were sufficient for three-state models. Different one-state systems and three-state systems are compared in Table 4.1.

system	1 state	3 states	difference
MFCC, 9 frames	39.9	35.6	4.3
MFCC39	37.7	32.8	**4.9**
MFCC39, 4 frames	34.1	29.9	4.2
simplified system (310 ms, 11 $DCTs$)	29.9	28.7	1.2
3 band TRAPs	29.2	25.8	3.4
LC-RC system	28.5	**24.4**	4.1

Table 4.1: *Comparison of 1-state and 3-state systems*

The three state systems are able to significantly reduce the phoneme error rate. The LCRC system profits about 4.1 % from the 3-state system. The simplified system has the smallest reduction.

4.2 Other architectures

All the previous experiments indicated that the clue to build a good recognizer based on HMM/ANN is the ability to focus the training algorithm on well defined coherent segments with as descriptive features as possible. Let us experiment with some more variants of the TRAP and the LC-RC systems.

4.2.1 How many bands in the TRAP multiband system are optimal?

If the number of joint bands is small, the band neural network does not have enough information for classification, the error rate is higher and the input pattern for merger is very difficult. If the number of joint bands is higher, the band neural network input patterns start to be difficult. A tradeoff must be found. The optimal number of joint bands is evaluated in Table 4.2. For wideband speech, the optimal number is 5. Another experiment showed that 3 is optimal for narrow band speech.

4.2.2 Split temporal context system (STC) with more blocks

The trajectory in feature space representing phoneme can be split into more than two parts and a generalization of the LC-RC system can be done (see Figure 4.1b). In this experiment the optimal

# bands per net	1	3	5	7	13
PER (%)	28.2	25.8	**24.8**	24.9	25.6

Table 4.2: *Optimal number of joint bands for band neural network in the 3 state multiband TRAPs system*

# blocks	1	2	3	5
PER (%)	26.8	24.4	24.2	**23.4**

Table 4.3: *Optimal number of blocks in 3-state split temporal context system*

number of parts is found. The input temporal vectors are split to 2, 3 and 5 parts. The Hamming windows are applied to all parts followed by dimensionality reduction to 11, 8, and 5 bases by DCT.

The tradeoff must be found even here. If the number of parts increases, the input pattern for merger also increases and starts to be difficult. The results can be seen in Table 4.3. The best number of blocks is 5. It may be even more, but this was not evaluated – the system starts to be slow and impractical.

4.2.3 Combination of both – split in temporal and split in frequency domain

The system is called ”2 x 2 system” – two temporal parts and two frequency parts (see Figure 4.1c). The system contains 5 neural networks (4 blocks and 1 merger). The preprocessing is similar to the preprocessing for the LC-RC system.

4.2.4 Comparison of the TRAP, STC and ”2x2” architecture

The architectures are compared in Table 4.4. The lowest PER is obtained by the 5 block STC system. But the PERs for the 5 band TRAPs and the ”2 x 2” systems are very close. This proved that both assumption – split in time and split in frequency – are helpful. It is not very important how the split is done. It is more important that the obtained patterns are easily learnable by the neural networks. The STC (LC-RC) system is used in later experiments because it needs less computer resources.

4.3 Tuning to the best performance

The STC with 5 blocks was taken and tuned to the best performance mainly by improved NN training: The scheduler for neural network learning rate was changed to use the *training set*. The scheduler halves the learning rate learning if the decrease in the frame error rate (FER) is less than 0.5% (the *cross-validation set* vas used before). The number of training epochs was fixed at 20.

Then, the numbers of hidden layer neurons in networks were increased from 500 to 800. I have seen that it was almost impossible to overtrain neural networks with 800 neurons in 20 epochs,

system	1 state	3 states
3 band TRAPs	29.2	25.9
5 band TRAPs	-	24.8
STC - 2 blocks (LC-RC)	28.5	24.4
STC - 5 blocks	-	**23.4**
2 x 2	-	24.1

Table 4.4: *Comparison of different time and/or frequency split neural network architectures.*

system	PER (%)
STC - 5 blocks	23.4
20 epochs in training	22.7
20 epochs in training + 800 neurons	22.1
+ CV part (18 minutes)	21.8
+ bigram LM	**21.5**[2]

Table 4.5: *Improvements to the 5-block STC system*

therefore the CV set was added to the training one. At the end, bigram language model[1] estimated (without any smoothing) on phonetic transcriptions of the training part was included. All described steps are summarized in Table 4.5.

4.4 Discussion

This chapter showed that the results can be significantly improved by a few easy and cheap tricks – finer representation of neural network outputs, introduction of more independence assumption to the neural network structure, more epochs in neural network training and a language model.

Also, few other structures of neural networks were studied. Although for example the tandem structure seems to be very perspective, it is not used later due to its higher complexity and more difficult training. It is rather a motivation for an investigation of different neural network structures.

[1] Known as phonotactic model in language recognition

[2] This correspond to the classification error rate 17.2%

Chapter 5

Conclusions

This work showed that it is possible to develop highly accurate phoneme recognizers on very low amount of training data. The accuracy comes from modelling of long temporal contexts for phonemes (few hundreds of milliseconds). The difficulty is the design of models for such large phoneme patterns. This thesis describes many techniques that allow to train neural networks for this purpose. The most important one is incorporation of some constraints coming from the task to the neural network structure. The possibility that the training algorithm will get stuck in local extreme is reduced. A hierarchical structure of neural networks was proposed for this purpose. The other techniques are dimensionality reduction of input patterns, windowing of the patterns or a finer representation of neural network outputs. Such designed phoneme recognizer with the split temporal context was integrated to a software package and it is now publically available on our web page[1].

A reviewer of one of my articles argued that ”the TIMIT was beaten to dead by this work”. It is impossible to study new promising techniques without coming to their limits and without having well trained classifiers. Although the phoneme error rate is already low (21.48 %), it is definitely not the final number, and even not for this unadapted system. Different normalization techniques, better language model, duration modelling or other complementary features can be applied. Then the system can by improved by speaker adaptation, speaker adaptive training, channel compensation and other techniques.

All the reported results here are phoneme recognition error rates. But a lower phoneme recognition error rate does not automatically mean lower word recognition error rate. It is always necessary to verify the advantage of new techniques on the final task. The relation between phoneme error rate, word error rate and language models will be studied in my future work.

[1] `http://speech.fit.vutbr.cz/en/software/phoneme-recognizer-based-long-temporal-context`

Bibliography

[1] G. Evermann, H.Y. Chan, M.J.F. Gales, B. Jia, X. Liu, K.C. Sim D. Mrva, P.C. Woodland L. Wang, and K. Yu. Development of the 2004 cu-htk eenglish cts systems using more than two thousand hours of data. In *Proceedings of Fall 2004 Rich Transcription Workshop*, Newyork, November 2004.

[2] F. Grezl. Combinations of trap based systems. In *Proc. Seventh International conference on Text, Speech and Dialogue*, pages 323–330, Brno, Czech Republic, 2004. Faculty of Informatics MU.

[3] F. Grezl. *TRAP-based Probabilistic Features for Automatic Speech Recognition*. PhD thesis, Brno University of Technology, 2007.

[4] H. Hermansky and P. Jain. Down-sampling speech representation in asr. In *Proc. Eurospeech*, volume 2, Hungary, Budapest, Sep 1999.

[5] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, Oct 1984.

[6] H. Hermansky and S. Sharma. Temporal patterns (traps) in asr of noisy speech. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Phoenix, Arizona, USA, Mar 1999.

[7] J. Černocky P. Schwarz, P. Matějka. Recognition of phoneme strings using trap technique. In *Proc. Interspeech*, pages 1–4, Geneve, Switzerland, 2003.

[8] S. van Vuuren and H. Hermansky. On the importance of components of modulation spectrum for speaker verification. In *Proc. International Conferences on Spoken Language Processing (ICSLP)*, volume 2, Australia, Sydney, 1998.

[9] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke. On using mlp features in lvcsr. In *Proc. International Conferences on Spoken Language Processing (ICSLP)*, Jeju Island, South Korea, Oct 2004.

Author

Petr Schwarz
`http://www.fit.vutbr.cz/˜schwarzp`

Petr Schwarz was born on July 11, 1977 in Jíčín, Czech republic. He received his Master's degree in Electrical Engineering from the Brno University of Technology in June 2001. He is postgradual student of Speech processing group at the Faculty of Information Technology (FIT), BUT since September 2001. He has been with the Anthropic speech processing group of Oregon Graduate Institute of Science and Technology, USA, from 10/2002 till 6/2003. Petr has actively participated in EU-projects SpeechDat-East, SpeeCon, M4 (Multimodal Meeting manager) and AMI (Augmented MultiParty Interaction) as well as in project ”Voice technologies for support of information society” sponsored by Grant Agency of Czech Republic (GACR). He is currently participating in EU-projects AMIDA (Augmented MultiParty Interaction with Distant Access), CareTaker, and in GACR project ”New trends in research and application of voice technology”. Petr Schwarz is author or co-author of more than 15 papers in journals and reviewed international conferences. He is member of IEEE and ISCA. His research interests include speech recognition, namely phoneme recognition based on novel feature extractions (temporal patterns) and neural networks. He has been active also in keyword-spotting, speaker and language identification and in real-time implementation of speech processing algorithms. Petr was a key member of teams successful in NIST LRE and NIST SRE evaluations.

Abstract

Techniques for automatic phoneme recognition from spoken speech are investigated. The goal is to extract as much information about phoneme from as long temporal context as possible. The Hidden Markov Model / Artificial Neural Network (HMM/ANN) hybrid system is used. At first, the Temporal Pattern (TRAP) system is implemented and compared to other systems based on conventional feature extraction techniques. The TRAP system is analyzed and simplified. Then a new Split Temporal Context (STC) system is proposed. The system reaches better results while the complexity was reduced. Then the system was improved using commonly used techniques such as three-state phoneme modelling and phonotactic language model. This system reaches 21.48 % phoneme error rate on the TIMIT database.