# Towards Lower Error Rates in Phoneme Recognition

Petr Schwarz, Pavel Matějka, and Jan Černocký

Brno University of Technology, Czech Republic
schwarzp|matejkap|cernocky@fit.vutbr.cz

**Abstract.** We investigate techniques for acoustic modeling in automatic recognition of context-independent phoneme strings from the TIMIT database. The baseline phoneme recognizer is based on TempoRAl Patterns (TRAP). This recognizer is simplified to shorten processing times and reduce computational requirements. More states per phoneme and bi-gram language models are incorporated into the system and evaluated. The question of insufficient amount of training data is discussed and the system is improved. All modifications lead to a faster system with about 23.6 % relative improvement over the baseline in phoneme error rate.

## 1 Introduction

Our goal is to develop a module which would be able to transcribe speech signals into strings of unconstrained acoustic units like phonemes and deliver these strings together with temporal labels. The system should work in tasks like keyword spotting, language/speaker identification or as a module in LVCSR. This article investigates mainly techniques for acoustic modeling. The TRAP based phoneme recognizer has shown good results [1], therefore this system was taken as a baseline. The TRAP-based system was simplified with the goal of increasing processing speed and reducing complexity. The influence of wider frequency band (16000 $Hz$ instead of 8000 $Hz$) was evaluated to keep track with previous experiments [1]. Then two classical approaches for better modeling – HMM (Hidden Markov Model) with more states and bi-gram language model – were incorporated into the system and evaluated. The main part of the work addresses the problem of insufficient amount of training data for acoustic modeling in systems with long temporal context, and tries to solve it. Two methods are introduced – weighting of importance of values in the temporal context and temporal context splitting.

## 2 Experimental systems

### 2.1 TRAP based system

Our experimental system is an HMM - Neural Network (HMM/NN) hybrid. Critical bands energies are obtained in the conventional way. Speech signal is

divided into 25 $ms$ long frames with 10 $ms$ shift. The Mel filter-bank is emulated by triangular weighting of FFT-derived short-term spectrum to obtain short-term critical-band logarithmic spectral densities. TRAP feature vector describes a segment of temporal evolution of critical band spectral densities within a single critical band. The central point is actual frame and there is equal number of frames in past and in future. That length can differ. Experiments showed that the optimal length for phoneme recognition is about 310 $ms$ [1]. This vector forms an input to a classifier. Outputs of the classifier are posterior probabilities of sub-word classes which we want to distinguish among. In our case, such classes are context-independent phonemes or their parts (states). Such classifier is applied in each critical band. The merger is another classifier and its function is to combine band classifier outputs into one. Both band classifiers and merger are neural nets. The described techniques yield phoneme probabilities for the center frame. These phoneme probabilities are then fed into a Viterbi decoder which produces phoneme strings. The system without the decoder is shown in figure 1.

One possible way to improve the TRAP based system is to add temporal vectors from neighboring bands at the input of band classifier [1, 6]. If the band classifier has input vector consisting of three temporal vectors, the system is called 3-band TRAP system.

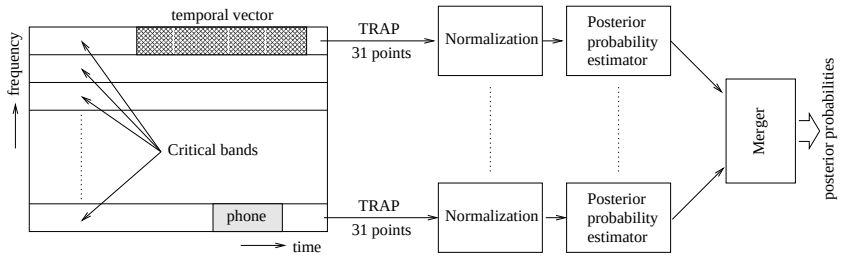


**Fig. 1.** *TRAP system*

### 2.2 Simplified system

The disadvantage of the system described above is its quite huge complexity. Usual two requests for real applications are shortest delay (or short processing time) and low computational requirements. Therefore we introduced a simplified version of the phoneme recognition system.

The system is shown in figure 2. As can be seen, band classifiers were replaced by a linear transform with dimensionality reduction. The PCA (Principal Component Analysis) was the first choice. During visual check of the PCA base components, these components were found to be very close to DCT (Discrete Cosine Transform), therefore the DCT is used further. The effect of simplification from PCA to DCT was evaluated and does not increase errors rates reported in this article by more than 0.5 %. It is necessary to note that PCA allows greater

reduction of feature vector dimensionality – from 31 to approximately 10 instead of 15 in case of DCT. Another modification is a window applied before DCT. Its purpose will be discussed later.

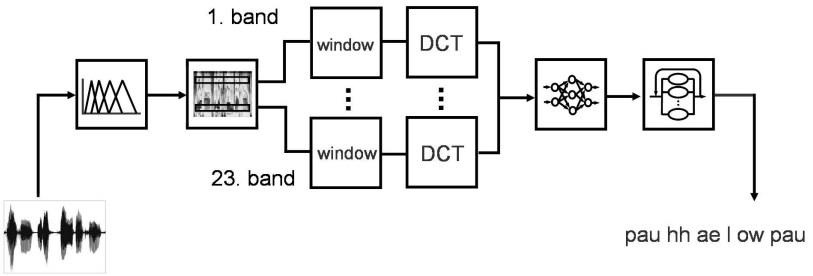


**Fig. 2.** *Simplified system – band classifiers were replaced by linear projections.*

## 3 Experimental setup

**Software** – a Quicknet tool from the SPRACHcore package [7], employing three layer perceptron with the softmax nonlinearity at the output, was used in all experiments. The decoder was written in our lab and implements classical Viterbi algorithm without any pruning.
**Phoneme set** – The phoneme set consists of 39 phonemes. It is very similar to the CMU/MIT phoneme set [2], but closures were merged with burst instead of with silence (bcl b $\rightarrow$ b). We believe it is more appropriate for features which use a longer temporal context such as TRAPs.
**Databases** – The TIMIT database was used in our experiment. All SA* records were removed as we felt that the phonetically identical sentences over all speakers in the database could bias the results. The database was divided into three parts – training (412 speakers), cross-validation (50 speakers), both form the original TIMIT training part, and test (168 speakers). The database was down-sampled from 16000 $Hz$ to 8000 $Hz$ for some experiments.
**Evaluation criteria** – Classifiers were trained on the training part of the database. In case of NN, the increase in classification error on the cross-validation part during training was used as a stopping criteria to avoid over-training. There is one ad hoc parameter in the system, the word (phoneme) insertion penalty, which has to be set. This constant was tuned to the equal number of inserted and deleted phonemes on the cross-validation part of the database. Results were evaluated on the test part of database. Sum of substitution, deletion and insertion errors – the phoneme error rate (PER) is reported. An optimal size of the neural net hidden layer was found for each experiment separately. Simple criteria – minimal phoneme error rate or negligible improvement in PER after the addition of new parameters – were used for this purpose.

## 4 Evaluation of classical modeling techniques

### 4.1 Baseline system, simplified system and 3-band system

Phoneme error rates are compared for all here mentioned system in Table 3. Our baseline is the one band TRAP system which works with speech records sampled at 16000 $Hz$. This system was simplified. The simplified version contains weighting of values in temporal context by Hamming window and dimensionality reduction after DCT to 15 coefficients for each band. 3 band TRAP system gave us better result than simplified system every time. The system models relations between three neighboring frequency bands, that in the simplified system is omitted.

### 4.2 16000 $Hz$ vs. 8000 $Hz$

In all experiments previously done with the TRAP based phoneme recognizer [1], the TIMIT database was down-sampled from 16000 $Hz$ to 8000 $Hz$ because of evaluating the system in mismatched condition where the target data was from a telephone channel. Now we are working with the wide band but wanted to evaluate the effect of down-sampling to 8000 $Hz$. The simplified system was trained at first using original records and then using down-sampled records. By the down-sampling, we loose 2.79 % of PER.

### 4.3 Hidden Markov Models with more states

Using more than one state in HMM per acoustic unit (phoneme) is one of the classical approaches to improve PER in automatic speech recognition systems. A speech segment corresponding to the acoustic unit is divided into more stationary parts that ensure better modeling. In our case, a phoneme recognition system based on Gaussian Mixture HMM and MFCC features was trained using the HTK toolkit [8]. Then, state transcriptions were generated using this system and neural nets were trained with classes corresponding to states. Coming up from one state to three states improved PER every time. Improvements are not equal and therefore this results are presented for each system separately. The improvement lies between 1.2 and 3.8 %.

### 4.4 Bi-gram Language Model

Our goal is to recognize unconstrained phoneme string, but many published results have the language model effect already included in and we wanted evaluate its influence. The language model was estimated from the training part of database. PER improvements seen from its utilization are almost consistent among all experiments and lie between 1 and 2 %.

## 5 Dealing with insufficient amount of training data

This experiment shows us how much data we need and whether it has sense to look for other resources or not. The training data was split into chunks half an hour long. The simplified recognizer was trained using one chunk, evaluated and then next chunk was added. The process was repeated for all chunks. The table 1 shows results. We are not so far in area of saturation with 2.5 hours of training data so we can conclude that adding more data would be beneficial.

| amount of training (hours) | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 |
|---|---|---|---|---|---|
| PER (%) | 46.13 | 42.26 | 39.86 | 39.23 | 37.92 |

**Table 1.** *Influence of number of training data on PER*

### 5.1 Motivations for new approaches

Many common techniques of speech parameterization like MFCC (Mel Frequency Cepstral Coefficients) and PLP (Perceptual Linear Prediction) use short time analysis. Our parameterization starts with this short term analysis but does not stop there – the information is extracted from adjacent frames. We have a block of subsequent mel-bank density vectors. Each vector represents one point in $n$-dimensional space, where $n$ is the length of the vector. All these points can be concatenated in time order, which represents a trajectory. Now let suppose each acoustic unit (phoneme) to be a part of this trajectory. The boundaries tell us places where we can start finding information about the phoneme in the trajectory and where to find the last information. Trajectory parts for two different acoustic units can overlap. This comes from the co-articulation effect. The phoneme may even be affected by a phoneme occurred much sooner or later than the first neighbors. Therefore, a longer trajectory part associated to an acoustic unit should be better for its classification.

We attempt to study the amounts of data available for training classifiers of trajectory parts as a function of the length of those parts. As simplification, consider the trajectory parts to have lengths in multiples of phonemes. Then the amounts are given by the numbers of n-grams[1]. Table 2 shows coverage of n-grams in the TIMIT test part. The most important column is the third, numbers in brackets – percentage of n-grams occurring in the test part but not in the training part. If we extract information from a trajectory part approximately as long as one phoneme, we are sure that we have seen all trajectory parts for all phonemes during training (first row). If the trajectory part is approximately as two phonemes long (second row), we have not seen 2.26 % of trajectory parts during training. This is still quite OK because even if each of those unseen trajectory parts generated an error, the PER would increases only by about

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

0.13 % (the non-seen trajectory parts occur less often in the test data). However, for trajectory parts of lengths 3 phonemes, non-seen trajectory parts can cause 7.6 % of recognition errors and so forth.

This gave us a basic feeling how the parameterization with long temporal context works, showed that a longer temporal context is better for modeling of the co-articulation effect but also depicted the problem with insufficient amount of training data. Simply, we can trust the classification less if the trajectory part is longer because we probably did not see this trajectory part during training. Next two paragraphs suggest how to deal with this problem.

| 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.** *Numbers of occurrence 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, error that would be caused by omitting not-seen N-grams in the decoder.*

### 5.2 Weighting values in the temporal context

We have shown that longer temporal trajectories are better for classification but that the boundaries of these trajectories might be less reliable. A simple way of delivering information about importance of values in the temporal context to the classifier (in our case neural net) is weighting. This can be done by a window applied on the temporal vectors before linear projection and dimension reduction (DCT) – figure 2. A few windows were examined and evaluated for minimal PER. The best window seems to be an exponential one where the first half is given by function $w(n) = 1.1^x$ and the second half is mirrored. For simplicity, the triangular window was used in all the following experiments. Note that it is not possible to apply the window without any post-processing (DCT) because the neural net would compensate for it.

### 5.3 Splitting the temporal context

In this approach, an assumption of independence of some values in the temporal context was done. Intuitively two values at the edges of trajectory part, which represent the investigated acoustic unit, are less dependent than two values closed to each other. In our case, the trajectory part was split into two smaller parts – left context part and right context part. A special classifier (again neural net) was trained for each part separately, the target units being again phonemes or states. An output of these classifiers was merged together by another neural net (figure 3). Now we can look at table 2 to imagine what has happened.

Let us suppose the original trajectory part (before split) was approximately as three phonemes long (3rd row). We did not see 18.83% of patterns from the test part of database during training. After splitting we moved one row up and just 2.26% patterns for each classifier was not seen. An evaluation of such system and comparison with others can be seen in table 3.

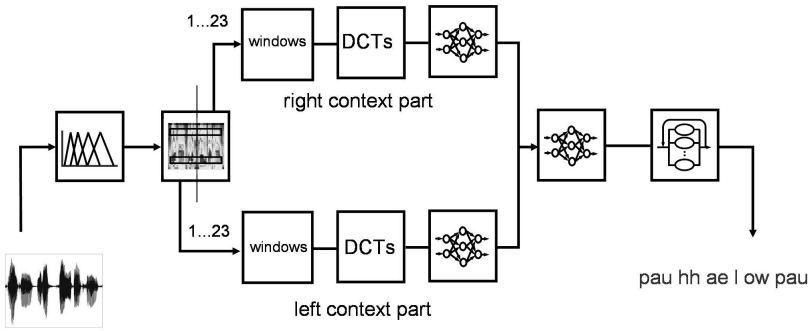


**Fig. 3.** *System with split left and right context parts.*

### 5.4 Summary of results

The system with split temporal context showed better results against baseline but its primary benefit comes in link with more than one state. For three states the improvement in PER is even 3.76 % against the one state system. Our best system includes weighting of values in temporal context, temporal context splitting, three states per acoustic unit and bi-gram language model. The best PER is 25.54 %. Till now the phoneme insertion penalty in the decoder was tuned to the minimum number of wrongly inserted an deleted phonemes on the cross-validation part of database. To fully gain from the PER measure, the optimization criteria for tuning the penalty was changed to minimal PER too. This reduces the PER about 1 % and leads to the final PER of 24.50 %.

## 6 Conclusion

The TRAP and 3-band TRAP based systems were evaluated on recognition of phoneme strings from the TIMIT database. Then the TRAP base system was simplified with the goal of shorting recognition time and reducing complexity. Classical approaches, which reduce phoneme error rates, like recognition from wider frequency band, HMM with more states or bi-gram language model were incorporated into the system and evaluated. Finally the problem of insufficient number of training data for long temporal contexts was addressed and two approaches to solve this problem were proposed. In the first one, the values in the temporal context are weighted prior to the linear transform. In the second

| | |
|---|---|
| **1 band TRAP system - baseline** | 33.44 |
| + bi-gram LM | |
| **Simplified system** | 31.64 |
| + 3 states | 30.74 |
| + bi-gram LM | 29.39 |
| **3 band TRAP system, 3 states** | 28.44 |
| + bi-gram LM | 27.37 |
| **Split left and right context** | |
| left context only | 37.30 |
| right context only | 39.56 |
| merged left and right contexts | 30.36 |
| + 3 states | 26.60 |
| + bi-gram LM | 25.54 |
| + max. accuracy | **24.50** |

**Table 3.** *Comparison of phoneme error rates. "max. accuracy" means that the criteria for tuning a phoneme insertion penalty (in decoder) was changed from equal number of wrongly inserted and deleted phonemes to the maximum accuracy.*

one, the temporal context is split into two parts and an independent classifier is trained for each of them. All these changes result in a faster system which improves the phoneme error rate of the baseline by more than 23.6 % relative.

## 7 Acknowledgments

This research has been partially supported by Grant Agency of Czech Republic under project No. 102/02/0124 and by EC project Multi-modal meeting manager (M4), No. IST-2001-34485. Jan Černocký was supported by post-doctoral grant of Grant Agency of Czech Republic No. GA102/02/D108.

## References

[1] P. Schwarz, P. Matějka and J. Černocký, "Recognition of Phoneme Strings using TRAP Technique", in Proc. Eurospeech 2003, Geneve, September 2003.
[2] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models", IEEE Transactions on Acoustics, Speech, and Signal Processing, 37(11):1641-1648, November 1989.
[3] A. Robinson," An application of recurrent nets to phone probability estimation", IEEE Transactions on Neural Networks, vol. 5, No. 3, 1994
[4] H. Bourlard and N. Morgan. "Connectionist speech recognition: A hybrid approach." Kluwer Academic Publishers, Boston, USA, 1994.
[5] H. Hermansky and S. Sharma, "Temporal Patterns (TRAPS) in ASR of Noisy Speech", in Proc. ICASSP'99, Phoenix, Arizona, USA, Mar, 1999
[6] P. Jain and H. Hermansky, "Beyond a single critical-band in TRAP based ASR", in Proc. Eurospeech'03, Geneve, Switzerland, September 2003.
[7] The SPRACHcore software packages, *www.icsi.berkeley.edu/ dpwe/projects/sprach/*
[8] HTK toolkit, *htk.eng.cam.ac.uk/*