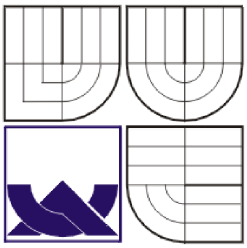
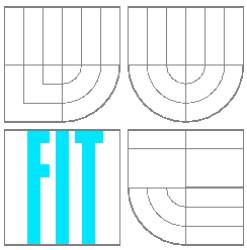


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

TRAP-BASED PROBABILISTIC FEATURES FOR AUTOMATIC SPEECH RECOGNITION

TITLE

DISERTAČNÍ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE
AUTHOR

ING. FRANTIŠEK GRÉZL

VEDOUCÍ PRÁCE
SUPERVISOR

DOC. DR. ING. JAN ČERNOCKÝ

BRNO 2007

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO
uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: František Grézl
Bytem: Bilsko 27, 783 22 Cholina
Narozen/a (datum a místo): 16. 3. 1977 Šternberk
(dále jen „autor")

a

2. Vysoké učdení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2, 612 66 Brno
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
Tomáš Hruška, prof. Ing., CSc.
(dále jen „nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - ■ disertační práce
   - □ diplomová práce
   - □ bakalářská práce
   - □ jiná práce, jejíž druh je specifikován jako .....

   (dále jen VŠKP nebo dílo)

Název VŠKP:	TRAP-based probabilistic features for automatic speech recognition
Vedoucí/ školitel VŠKP:	Doc. Dr. Ing. Jan Černocký
Ústav:	Ústav počítačové grafiky a multimédií
Datum obhajoby VŠKP:	

   VŠKP odevzdal autor nabyvateli v[1]:
   - ■ tištěné formě – počet exemplářů 3
   - □ elektronické formě – počet exemplářů

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

[1] hodící se zaškrtněte

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.

2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.

3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - ■ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy
   - (z důvodu utajení v něm obsažených informací)

4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.

2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.

3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.

4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 27. 7. 2007

.....	.....
Nabyvatel	Autor

Abstract

For automatic speech recognition and many other applications, feature extraction transforms the signal to a feature vector that is modelled by the subsequent stochastic classifiers. The quality of feature extraction has direct effect on the quality of the classification. We are investigating the probabilistic features based on temporal evolution of critical band spectrogram. Our work has witnessed the evolution of these features from being poor relative of standard cepstral features, through being a useful complementary component, to current state, where they outperformed the cepstral features and became inseparable part of state-of-the-art systems.

We first worked on various alternations of derivation of probabilistic features from temporal trajectories such as dimensionality reduction of temporal vectors, concatenation of temporal trajectories from adjoining critical bands and dimensionality reduction of the resulting vector. We proposed modification of critical band spectrogram prior to the extraction of the critical band trajectory. This simple operation helps system to focus on one aspect of original critical band spectrogram only. Further, we have investigated combinations of systems based on differently modified spectrograms. Different scenarios of this combination are examined too. This study is done on small vocabulary digit recognition experiment.

The systems with good performance are introduced to noisy speech recognition task on AURORA 2 database. These experiments allow analysing of behaviour of systems in noisy conditions. Finally, proposed features are introduced to large vocabulary continuous speech recognition task. The features are combined with cepstral features for this task and the evaluation of systems performances is conducted. We observed that TRAP-based probabilistic features can significantly improve the performance of automatic speech recognition.

Keywords

Temporal trajectories, temporal processing, band merging, spectrogram modification, system combination, multi-stream combination, noise robustness, LVCSR system, probabilistic features.

Bibliographic citation

František Grézl: TRAP-based probabilistic features for automatic speech recognition, **Doctoral thesis**, **Brno**, **Brno University of Technology, Faculty of Information Technology**, 2007

Abstrakt

Pro automatické rozpoznávání řečí a mnoho jiných aplikací, parametriace transformuje signál do příznakového vektoru, který je modelován následným stochastickým klasifikátorem. Kvalita parametrizace má přímý vliv na kvalitu klasifikace. Zabýváme se pravděpodobnostními příznaky založenými na tvaru průběhu úzkopásmové energie v čase. Náše práce dokumentuje vývoj těchto příznaků od slabých příbuzných standardnách cepstrálních příznaků, přes stav, kdy byly užitečným doplňkem do současného stavu, kdy překonali cepstrální příznaky a staly se neoddělitelnou součástí nejlepších systémů.

Nejdříve pracujeme s pravděpodobnostními příznaky které jsme získali různým způsobem z časových průběhů jako například redukcí dimenzionality, spojením několika časových průběhů ze sousedních pásem a zredukováním dimanzionality výsledného vektoru. Dále navrhujeme modifikaci pásmového spektrogramu před získáním průběhů z jeho pásem. Tato jednoduchá operace zaměří systém pouze na jednu část informace obsažené v původním pásmovém spektrogramu. Zkoumáme i kombinaci takto modifikovaných systémů a navrhli a otestovali jsme několik kombinačních technik. Tato studie je provedena na malém rozpoznávači číslovek.

Systémy, které pracovali nejlépe na našem malém experimentu jsme dále použili pro rozpoznávání zašuměné řeči z databéze AURORA 2. Tyto experimenty nám umožnily analyzovat chování systému v šumových podmínkách. Závěrem jsou navrhované příznaky otestovány na rozpoznávání spojité řeči s velkým slovníkem. Pro tuto úlohu jsou námi navrhované příznaky testované ve spojení se standardními příznaky. Po provedení testů můžeme říci, že příznaky založené na tvaru průběhu úzkopásmové energie mohou významě zlepšit funkčnost systémů pro automatické rozpoznáváví řeči.

Klíčová slova

Časové trajektorie, časové zpracování, spojování pásem, modifikace spektrogramu, kombinace systémů, vícesystémová kombinace, šumová odolnost, LVCSR systém, pravděpodobnostní příznaky.

Bibliografická citace

František Grézl: TRAP-based probabilistic features for automatic speech recognition, **Disertační práce**, **Brno**, **Vysoké Učení Technické v Brně, Fakulta informačních technologií**, 2007

Prohlášení

Prohlašuji, že jsem tuto disertační prácí vypracoval samostatně pod vedením Doc. Dr. Ing. Jana Černockého.

Další informace mi poskytli Hynek Hermansky a Pratibha Jain ohledně základů příznaků založených na časových trajektoriích a rozpoznávače číslovek. Martin Karafiát mi podal potřebné informace k experimentům na Meetingových datech. Andreas Stolcke, Nelson Morgan, Barry Chen a Qifeng Zhu mi podali informace související s rozpoznávacím systémem používaným na ICSI.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Acknowledgements

On this special occasion of finishing my Ph.D. thesis I would like to thank all the people who helped me with the research and also filled the time with unforgettable memories. On the first place I would like to thank to my adviser Honza Černocký who admitted me as his student and finally stopped asking, when I'm going to finish this thesis. My thanks also belong to Hynek Hermansky, a leader of former Anthropic Signal Processing (ASP) group at Oregon Graduate Institute (OGI), who invited me for internship in his group and whose guidance give direction to my work. I would also like to thank to all ASP group members, Pratibha Jain, Sunil Sivadas, Sachin Kajarekar and Andre Adami, for their advices, comments and discussions and also for all the fun we had inside and outside the office. My thanks belongs also to Petr Fousek, my colleague from internship at IDIAP, for the lunch-time discussions and delicious cooking. I would like to thank to Hemant Misra and Hamed Ketabdar, members of IDIAP speech group, for their insights too. Many thanks belongs also to members of "Novel Approaches" subgroup of ICSI speech group I spent a very fruitful five months with: Barry Chen and Qifeng Zhu, who introduced me into CTS task at ICSI and provided me with necessary information to be able to run experiments, Andreas Stolcke who always helped me out with the DECIPHER system though he basically had no time, David Gelbart for his valuable comments, Mark Ferras and Xavi Anguera for making more fun in the place, and Nelson Morgan and Barbara Peskin, leaders of the speech group, who made the ICSI speech group a pleasant place to work. I would also like to thank to members of my "base" group, Speech@FIT, Lukáš Burget, Petr Motlíček, Pavel Matějka, Petr Schwarz, Martin Karafiát and Tomáš Kašpárek for their help and support during my work. Special thanks belongs to Martin, our LVCSR guru, who helped me with the meeting speech recognition and Tomáš who is maintaining our computer systems. Thanks belongs also to Igor Szöke, Ondra Glembek, Miso Fapšo, Jiří Kopecký, Kamil Chalupniček and Valja Hubeika for being around. My great thanks belongs to my parents, for their support and patience with my never-ending studies. Thanks also to my friend Hanka, who waited for my returns from my frequent internships and late returns from office. I also would like to thank to all my friends who remained in contact with me.

Thank you

Contents

List of Figures

List of Tables

Chapter 1

Introduction to speech recognition

Speech is a natural way of communication. The goal of Automatic Speech Recognition (ASR) is to make this mode of communication available also for human-machine interaction. Although automatic speech recognition made a great progress since "Radio REX" in 1914[1], it still does not have the ability of systems we can see in sci-fi movies where people communicate with computers through speech.

Today's systems are obligated to its high performance to task and environment specific design. The performance of systems degrades in adverse conditions as shown in [57]. These degrading conditions origin from two major sources of speech signal variability: environmental variations and speaker variations.

Environmental variations include sounds other than speaker speech picked up by the microphone, i.e. noise from the room equipment, noises from street or other people speech. This kind of variation is called background noise. Other kind of environmental variation is the reverberation which is caused by the reflections of sound waves from the surfaces around the speaker. The third important kind of environmental variation is caused by the recording channel, i.e. technical parameters of the microphone, transmission and recording equipment.

Speaker variation occurs both across different speakers and within one speaker at different times. The within-speaker variations occurs when a speaker speaks at different rates, uses formal or slang vocabulary depending on the audience, or the voice can change when the speaker is tired, sick or overwhelmed by some emotions. The across-speaker variability occurs in the pitch of the voice, rhythm and accent of the voice.

A good speaker independent ASR system has to deal with all the variations. Speech recognition is a complex problem consisting of several parts. Each part is addressed by different techniques, based on different knowledge and data.

1.1 Bases of ASR

A typical ASR system is today based on statistical pattern recognition [74] and its goal is to find the most likely sequence of words given a set of input patterns (observations) and model parameters. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a sequence of N observation vectors, *feature vectors*. Let $\mathbf{W} = \{word_1, word_2, \ldots, word_M\}$ be a sequence of M words. The ASR system outputs such word sequence $\overline{\mathbf{W}}$ which maximizes the likelihood:

$$\overline{\mathbf{W}} = \arg\max_{W} P(\mathbf{W}|\mathbf{X}, \mathbf{\Theta}) \tag{1.1}$$

where $\mathbf{\Theta}$ is a set of all model parameters. Instead of building overall model $P(\mathbf{W}|\mathbf{X}, \mathbf{\Theta})$, we can factor this model into smaller models. First, we can split the words into distinct sounds. The phonemes[2]

[1] Radio REX – a toy in which a dog jumps out of his hut when his name is shouted.

[2] A phoneme is the theoretical representation of a sound without reference to its position in a word or phrase. Phonemes are not the physical segments themselves, but mental abstractions of them – phoneme is a group of sounds that the

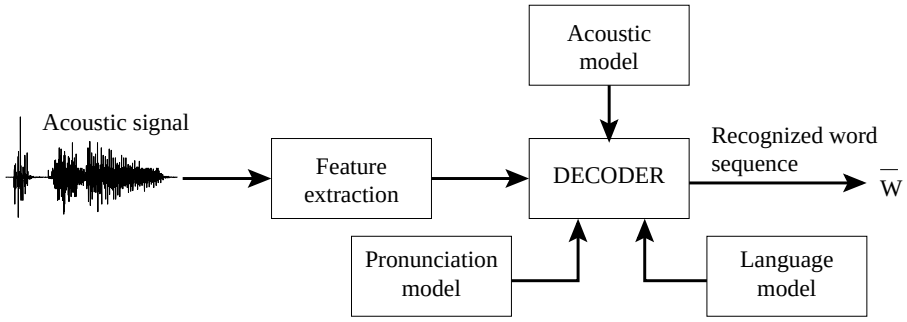


Figure 1.1: Block diagram of ASR system

are the most commonly used sub-word units. Let $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_K\}$ be a set of phonemes which can fully describe any word from $\mathbf{W}$. Thus the word sequence $\mathbf{W}$ in Eq. 1.1 can be replaced by all possible sequences of phonemes $\mathbf{Q}$ which together form the word sequence $\mathbf{W}$:

$$\overline{\mathbf{W}} = \arg\max_{W} \sum_{Q} P(\mathbf{W}, \mathbf{Q}|\mathbf{X}, \mathbf{\Theta}) \tag{1.2}$$

By using Bayes rule we can further obtain:

$$\overline{\mathbf{W}} = \arg\max_{W} \sum_{Q} \frac{P(\mathbf{X}|\mathbf{W}, \mathbf{Q}, \mathbf{\Theta}) P(\mathbf{W}, \mathbf{Q}|\mathbf{\Theta})}{P(\mathbf{X}|\mathbf{\Theta})} \tag{1.3}$$

Note, that the term in the denominator $P(\mathbf{X}|\mathbf{\Theta})$ is constant for all word sequences $\mathbf{W}$. Thus we can drop this term in maximization. Further we can factor the joint likelihood $P(\mathbf{W}, \mathbf{Q}|\mathbf{\Theta})$ and we obtain:

$$\overline{\mathbf{W}} = \arg\max_{W} \sum_{Q} P(\mathbf{X}|\mathbf{W}, \mathbf{Q}, \mathbf{\Theta}) P(\mathbf{Q}|\mathbf{W}, \mathbf{\Theta}) P(\mathbf{W}|\mathbf{\Theta}) \tag{1.4}$$

Finally, we assume conditional independence of observation sequence $\mathbf{X}$ on the word sequence $\mathbf{W}$ and we let it depend only on the phone sequence $\mathbf{Q}$. We further divide the set of parameters $\mathbf{\Theta}$ into parts each of which affects only the likelihood term it is contained in:

$$\overline{\mathbf{W}} = \arg\max_{W} \sum_{Q} P(\mathbf{X}|\mathbf{Q}, \mathbf{\Theta}_{AM}) P(\mathbf{Q}|\mathbf{W}, \mathbf{\Theta}_{PM}) P(\mathbf{W}|\mathbf{\Theta}_{LM}) \tag{1.5}$$

The three likelihood models in Eq. 1.5 are:

Acoustic model $P(\mathbf{X}|\mathbf{Q}, \mathbf{\Theta}_{AM})$ which models the probability of a observation sequence given a phoneme sequence.

Pronunciation model $P(\mathbf{Q}|\mathbf{W}, \mathbf{\Theta}_{PM})$ which tells us how likely is a sequence of phonemes given a sequence of words. The pronunciation model is also called "pronunciation dictionary" or only "dictionary".

Language model $P(\mathbf{W}|\mathbf{\Theta}_{LM})$ which gives the probability of a given word sequence.

These models together with feature extraction define major subdivision of ASR research. The block diagram of the ASR system is shown in Fig 1.1. Now, we will further describe each block.

speakers think of as being categorically the same.

1.1.1 Feature extraction

Though we did not speak about the feature extraction much so far, it is a very important part of the ASR system and the rest of this work will be only about the feature extraction. The acoustic modelling and mainly the relationship of models to underlying speech depends on the quality of the features derived from acoustic signal. The quality of features can be regarded as their ability to capture information related to the task – speech recognition. Thus we want to have features which pick up only the information about the **message** contained in speech from incoming acoustic signal and suppress all other information caused by all the degrading conditions discussed in the beginning of this chapter.

The main reasons why the feature extraction is needed and why the acoustic signal is not used directly are:

- reduction of dimensionality and redundancy
- removal of unwanted information such as information about the environment, speaker and transmission channel

1.1.1.1 Standard feature extraction

The evolution of the feature extraction was inspired by the knowledge of speech production and perception which is largely incorporated in extraction of two most popular features: Mel-Frequency Cepstral Coefficients (MFCC) [60] and Perceptual Linear Predictive coefficients (PLP) [33]. These features are standard in speech recognition and consist of the following steps:

1. **Signal preprocessing** – This step is applied optionally on the signal to increase the quality of the recording. Techniques like noise suppression or echo cancellation can be used here. The signal processing commonly used in ASR is "pre-emphasis" – high-pass filtering which increases the energy of higher frequency components.

2. **Segmentation** – Acoustic signal is divided into segments which can be regarded stationary. The typical duration of the segment is 25 ms. To preserve information about time evolution of speech signal, segments are taken with some overlap – typically 15 ms. The classifiers generally assume that their input is a sequence of discrete parameter vectors where each parameter vector represents just one such segment – a frame.

3. **Spectrum computation** – Short time Fourier power spectrum is computed from each frame.

4. **Auditory-like modifications** – Modifications inspired by physiological and psychological findings about human perception of loudness and different sensitivity of different frequencies are performed on spectra of each speech frame.

   The power spectrum is integrated into several frequency bins equally spaced on non-linear (Bark or Mel) frequency axis. These bins capture spectral energy only in given frequency band. Frequency band covered by one bin is called "critical band". The output critical band energies are further modified (log or cube-root) to simulate human perception of loudness.

5. **Decorrelation** – The decorrelation of critical band energies is done by techniques used for spectral analysis – Discrete Cosine Transform (DCT) in case of MFCC and linear prediction followed by the spectral conversion for PLP. Such transformed domain is called "cepstral" domain.

   Another purpose of this processing step is the dimensionality reduction. It is done by taking only several first (usually 13) cepstral coefficients.

6. **Derivatives** – Feature vectors are usually completed by first and second order derivatives of their time trajectories (delta and acceleration coefficients). These coefficients describe time evolution of the feature over approximately 100 ms.

The relative spectral (RASTA) technique [37] adds more sophisticated auditory-like modifications into the process of feature extraction and can be used in both MFCC and PLP feature extraction. This technique filters trajectory of critical band energy by a band-pass filter. The filter attenuates slow changes and DC component which may be caused by slow changes or spectral tilt in the transmission channel characteristic, and fast changes which may be caused by noise. Frequency band, which is characteristic for the speech, will pass through the filter.

There are other improvements and modifications suggested in publications for both feature extractions (e.g. [62, 40]), but these usually fail to make it into broad public and remain active only in the research group developing them.

1.1.1.2 Novel feature extraction

Novel features are usually based on different processing than standard features and try to bring new information into speech recognition system. These features can be used alone or in combination with the standard features. To give some examples, the "Phase Autocorrelation" features can be found in [43], "Spectral entropy" features in [64]. The "prosodic" features [52] are bringing the linguistics information about intonation, stress and rhythm into speech recognition system. These features are successful in speaker recognition [2, 79] and they are making their way into ASR [80].

More popular novel features are "probabilistic" features. These features are class probabilities transformed to a form suitable for the following acoustic model. They were first introduced in so called TANDEM ASR [34] where output of one classifier creates input to the second classifier. Although the outputs of the first classifier are now considered as features, the classifier has to generate its outputs based on some features too. And again, we can use standard features or some other kind of novel features as input for this classifier.

The TempoRAl Patterns (TRAP) features are one of the features suitable for TANDEM ASR. These features are basically temporal trajectories of critical band energies, where information from each critical band is processed independently [78].

1.1.2 Acoustic model

The acoustic model models the smallest parts of the speech – distinct sounds – phonemes. The state-of-the-art acoustic models use Hidden Markov Models (HMM) – probabilistic finite state machines [5, 74]. Each model represents one phoneme or a phoneme in specific context of preceding and following phoneme – context dependent phoneme.

The phoneme model consists of several states which represent portions of the phoneme. Hidden Markov models used in speech recognition are mostly simple left-to-right models without state skipping. Within a model, there are probabilities of transition from current state to the next one, denoted a_{ij}. Also, each emitting state is associated with a function giving the likelihood of emitting a feature (observation) vector $\mathbf{x}$ denoted $b(\mathbf{x})$. The scheme of one model with three emitting states is depicted in Fig. 1.2.

We assume that the feature sequence $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$ was generated by the Markov Model $\mathbf{Q}$. Each time $t = 1 \ldots 5$, a state S_i was entered and feature vector $\mathbf{x}_t$ was generated. The joint probability that the vector sequence $\mathbf{X}$ was generated by the model Q moving through the state sequence $\mathbf{S} = \{S_1, S_1, S_2, S_3, S_3\}$ (according to the Fig 1.2) is calculated simply as a product of the transition probabilities a_{ij} and the emitting likelihoods $b_i(x_t)$. For the case shown in Fig 1.2:

$$P(\mathbf{X}|Q, \mathbf{S}) = b_1(\mathbf{x}_1)a_{11}b_1(\mathbf{x}_2)a_{12}b_2(\mathbf{x}_3)a_{23}b_3(\mathbf{x}_4)a_{33}b_3(\mathbf{x}_5)a_{34} \tag{1.6}$$

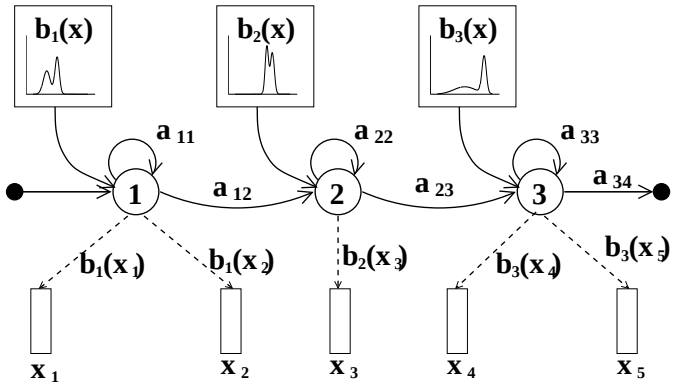


Figure 1.2: Scheme of Markov Model with three emitting states

However, in practice, only the vector sequence $\mathbf{X}$ is known and the state sequence $\mathbf{S}$ is hidden, thus the word "hidden" is associated with these models. Therefore the likelihood is computed over all possible state sequences $\mathbf{S}$ and the state sequence with the best likelihood is identified to be the one that generated the feature sequence $\mathbf{X}$.

$$P(\mathbf{X}|Q) = \max_{\mathbf{S}} \left(\prod_{t=1}^{T} b_{S_{(t)}}(\mathbf{x}_t)\ a_{S_{(t)}S_{(t+1)}} \right) \tag{1.7}$$

where $a_{S_{(t)}S_{(t+1)}}$ is the probability of transition from a state occupied in time t to state occupied in time $t+1$ and $b_{S_{(t)}}(\mathbf{x}_t)$ is the likelihood that the state occupied at time t emits vector $\mathbf{x}_t$. Note, that the process has to exit the model at time T to provide valid likelihood.

Most HMM systems represent the emitting likelihood $b(\mathbf{x})$ by Gaussian mixture model (GMM) of probability densities of feature vectors $\mathbf{x}$. The formula for computing $b_j(\mathbf{x}_t)$ is then

$$b_j(\mathbf{x}_t) = \sum_{i=1}^{M} c_i N(\mathbf{x}_t, \boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_{ji}), \tag{1.8}$$

where M is the number of mixture components, c_i is the weight of the i-th component and $N(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the output value of a multivariate Gaussian with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, that is

$$N(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})} \tag{1.9}$$

Instead of full covariance matrix, only its diagonal vector which represents variances of feature coefficients is mostly used. Other matrix elements are assumed to be zero (this supposes that individual elements of the feature vector are not correlated, which is guaranteed by the decorrelation step in the feature extraction).

For training of acoustic model, we need a large amount of (parametrized) training data. These data should be transcribed on the phoneme level. Then the training algorithm will accumulate statistics for each state and estimate the corresponding parameters of Gaussian mixture model.

But usually, we have only audio data and corresponding word transcription. The phoneme level transcription is then created according to the pronunciation dictionary. If the dictionary does not correspond to pronunciations actually spoken, the statistics are accumulated from wrong sounds which impairs the quality of the acoustic model.

For the training, the features must contain the information about the underlying sound in such details that creation of distinct Gaussian models is possible for each state. Otherwise the training will result in very poor models or would not even be possible.

	word	probability of the pronunciation variant	phoneme transcription
similar words	POTATO	0.640	p oh th ah th ow
	POTATO	0.360	p ax t ey t ow
	SUBDIRECTORY	0.530	s ah b d ax r eh k t axr iy
	SUBDIRECTORY	0.470	s ah b d ay r eh k t axr iy
distinct words	GOING TO	0.700	g ow ih ng th ow
	GOING TO	0.300	g ao n ax
	ZERO	0.600	z iy r ow
	ZERO	0.400	ow
multiwords	THERE-WILL	0.470	dh ey r w ih l
	THERE-WILL	0.5300	dh ey r l

Table 1.1: The examples of pronunciation model entries

The quality of the acoustic model depends on quality (discriminability) of input features and quality of pronunciation dictionary. It influences all the recognition process as the phoneme acoustic models are basic construction units.

1.1.3 Pronunciation model

The pronunciation model (pronunciation dictionary) governs the concatenation of basic phoneme models into the words. The pronunciations can be found in dictionaries or they are defined by rules. Each pronunciation variant of a word can have its probability reflecting the relative use of the given pronunciation.

The pronunciation of the word may differ from its dictionary form only in phonemes which are acoustically similar or it may differ more, due to a regional dialect or in case the speaker is non-native.

It is also possible that one word has completely different pronunciations. This can happen when functional words or sign names are different but their meaning is the same. Also, a slang variant of a literary word can be quite different. If we are interested in meaning of the word only and not in its exact spoken form, these pronunciations belong to one word.

In spontaneous speech, word can also have a different pronunciation form in some context. Contraction and reduction phenomena happen specially in words in frequent phrases such as greetings or usual questions. Such phrases are called "multiwords" and they are studied in [24].

The example of pronunciation model entries is given in Tab. 1.1. The pronunciation model is usually created by a linguist who has to listen to the spoken terms and decide on the phoneme transcription. The quality of the model affects the recognition of individual words – words with wrong or insufficient pronunciation transcription will be poorly recognized. It also influences the training of acoustic model since the phoneme transcriptions are generated from word ones as mentioned above.

1.1.4 Language model

Language models estimate the probability of a word sequence, $\hat{P}(w_1, w_2, \ldots, w_m)$ that is, they evaluate $P(\mathbf{W}|\mathbf{\Theta}_{LM})$ as defined in Eq. 1.5. The probability $\hat{P}(w_1, w_2, \ldots, w_m)$ can be decomposed as a product of conditional probabilities:

$$\hat{P}(w_1, w_2, \ldots, w_m) = \prod_{i=1}^{m} \hat{P}(w_i|w_1, \ldots, w_{i-1}) \tag{1.10}$$

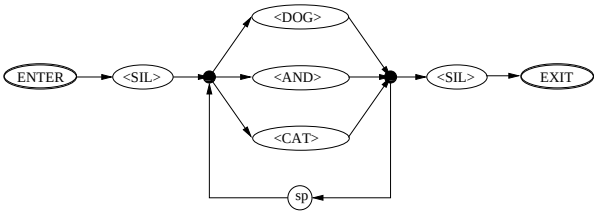


Figure 1.3: The word recognition net for continuous speech recognition.

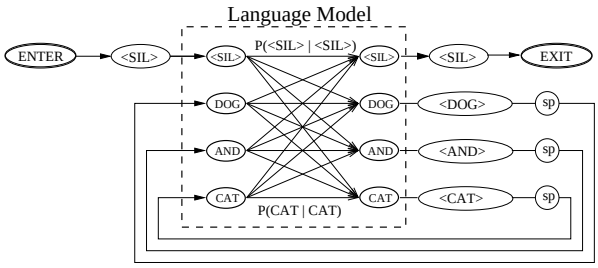


Figure 1.4: Expansion of recognition net by language model.

This equation presents an opportunity for approximating $P(\mathbf{W}|\mathbf{\Theta}_{LM})$ by limiting the context:

$$\hat{P}(w_1, w_2, \ldots, w_m) \simeq \prod_{i=1}^{m} \hat{P}(w_i|w_{i-n+1}, \ldots, w_{i-1}) \tag{1.11}$$

for some $n > 1$. If language is assumed to be ergodic – that is, it has the property that the probability of any state can be estimated from a long enough history independent of the starting conditions – then for sufficiently high n Eq. 1.11 is exact. However, due to practical reasons, values of n in the range of 1 to 4 inclusive are typically used. Models using contiguous but limited context in this way are usually referred to as *n-gram* language models, and the conditional context component of the probability ($w_{i-n+1}, \ldots, w_{i-1}$ in Eq. 1.11) is referred to as the *history*.

Estimates of probabilities in n-gram models are commonly based on maximum likelihood estimates – that is, by counting events in context on some given training text:

$$\hat{P}(w_i|w_{i-n+1}, \ldots, w_{i-1}) = \frac{C(w_{i-n+1}, \ldots, w_i)}{C(w_{i-n+1}, \ldots, w_{i-1})} \tag{1.12}$$

where $C(.)$ is the count of a given word sequence in the training text. The Eq. 1.12 covers only word sequences $\mathbf{W}$ which appear in the training data. To avoid the zero probability of unseen or infrequent word sequences the *back off* or *discounting* techniques are employed while training the LM [48].

To train language model, we need to count the occurrences of word sequences of certain length. It is obvious that the longer the desired word sequence is, the more data are needed for sufficient number of counts. The language model is mostly obtained from text data as there is only limited amount of transcribed speech data.

This naturally brings the problem of disagreement between speaking and writing style. To overcome it, speech transcriptions are added to text data with a bigger weight. The coverage of test data by the language model represents LM quality with respect to this test data.

The language model increases the probability of recognition of more likely sequences of the words. This helps to overcome problems with mispronounced words where acoustic model would prefer other phoneme then the one occurring in the correct word and the pronunciation model would choose word which is acoustically closer to the actual pronunciation. But this property of language model hurts the recognition if language model does not fit the speaking style or if words appear in unusual context.

1.1.5 Decoder

The decoder takes the partial models and creates a compound recognition net as follows:

- Language model may be thought of as network giving probability of word sequences.
- Words from language model are expanded according to the dictionary to chains of phonemes. Multiple pronunciations create parallel branches for given word.

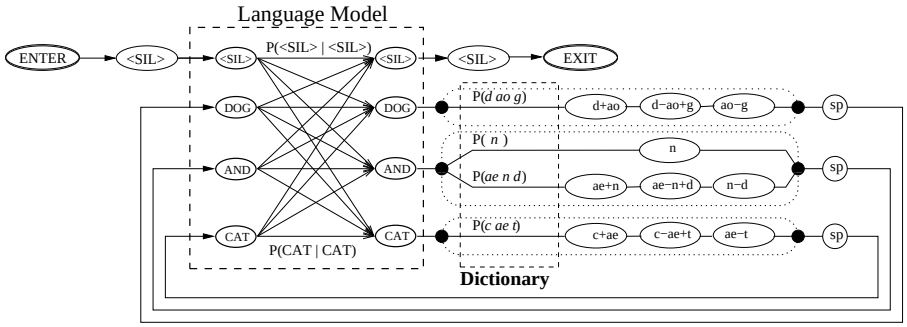


Figure 1.5: Expansion of word recognition net with context dependent phoneme models.

- Phoneme models are expanded by acoustic model to sequence of states. In case of context dependent acoustic models, the network has to be expanded with respect to the phoneme context.

The decoder task is to perform all these expansions, to create the recognition network – compound HMM — and to solve Eq. 1.5. It is usually done by Viterbi algorithm [91, 26].

As an example, we can take a simple recognizer of three words: "CAT", "DOG", "AND". A net with three parallel branches is created, each branch represents a model of one word. The silence models are added at the beginning and end of the net. To generate continuous speech recognition net, a loop back from the end node to a start one through the short-pause model is added. This word recognition net is shown in Fig. 1.3.

A system with this recognition net can output an arbitrary combination of the words from the dictionary. By adding language model, the transitions from one word to another are weighted, giving preferences to more probable word sequences. The extension of recognition net in Fig 1.3 by bigram language model (probability of the current word depends on the preceding word) is shown in Fig. 1.4.

The word nodes are then expanded to phonemes according to dictionary. Assume, that the pronunciation of the word "AND" will have two variants: one full *ae n d* and other reduced to *n* only. Then, two branches appear for word "AND" with different phoneme models on them. Further, phonemes are expanded by the context dependent phoneme models. Since in continuous speech recognition, it is not known what word precedes or follows a given word, the phoneme models on the word boundary have only one side context. The resulting recognition net is shown in Fig. 1.5. The probability of pronunciation variant is given by dictionary (dashed line). The expansion of words to context dependent phonemes is in dotted areas. This net can by further expanded into individual states.

1.2 Importance of feature extraction

Although there are other information sources than the actual speech signal, incorporated in pronunciation dictionary and language model, the information from speech is essential for speech recognition. If some useful information form speech is lost in the feature extraction process, it is not possible to recover it in further processing. That is why significant efforts are devoted to feature extraction. Also, the complex information merging, that takes place during decoding, can perform reliably only when acoustic models provide relevant and discriminative information. Features which capture more of the useful information contained in speech or represent the information in better way (for subsequent modelling) increase the performance of ASR system. Search for better features is a natural way towards more reliable speech recognition system and this work is devoted to it.

Traditionally, the feature extraction is based on the knowledge, intuition and belief of researchers. As mentioned in section 1.1.1, the knowledge of speech production and perception was used in feature extraction design. However, some processing steps were based only on rough optimization or set ad-

hoc. The two commonly used features (MFCC and PLP) contain processing steps which do similar thing, but the implementation techniques differ and the benefit from these differences is not evident. There are many modifications of the standard features which improve the recognition performance on some task. For example, in [40] authors tried to improve PLP features by tuning and enhancing individual processing steps. Similarly, attempts to train feature extraction – or at least some parts of it – on the data were made. These approaches were motivated by the fact, that the rest of ASR system is also trained on the data. The design of data-driven RASTA-like filters is shown in [88], the data-driven critical band filter-bank was obtained in [11]. But this direction of improving standard features by slight modifications is limited by the the framework of the original features computation.

The novel features offer much larger space for research, but often do not reach the performance of the standard features[3]. That is why they are often used in combination with the standard features. If the two features are going to be combined, it is useful to find such novel features, which contain complementary information to the one contained in standard ones. Such features appear to be probabilistic features based on TempoRAl Pattern – TRAP – processing [78]. The TRAP processing, in opposite to standard features, extracts information first in time domain and then combines it over frequency. This work deals with novel TRAP-based probabilistic features, their improvement and combination with standard features.

1.3 Scope of chapters

This work is organised as follows:

Chapter 2 gives motivation for TempoRAl Pattern – TRAP – processing from both, technical and theoretical points of view. The background leading to the use of narrow-frequency long-time representation is introduced. The properties and advantages of probabilistic features are also introduced.

The detailed description of TRAP-based probabilistic features, with possible processing of temporal patterns, is given next. The chapter is closed by description of evaluation and analysis of TRAP processing.

Chapter 3 describes in details experimental setup called "Stories-Numbers" – a digit recognition task used for evaluation of different TRAP-based probabilistic features. This task was used by several other researchers and direct comparison of results is possible. This task has fast experiment turnaround allowing quick testing of various ideas.

This chapter also describes the neural net training procedure which is the same for different experimental setups introduced later in this work.

Chapter 4 studies the performance of TRAP-based probabilistic features obtained with different processing of temporal pattern. Then, the merging of information from several frequency bands prior to merging is examined. Along with band merging, different processing of TRAP vectors is tested.

Chapter 5 introduces the spectrogram modification technique. Different modifying operators are used to obtain different modified spectrograms. The TRAP-based probabilistic feature extraction is then applied on top of these modified spectrograms. Different processings of TRAP vectors are also evaluated in this scenario.

Chapter 6 examines possible combination of TRAP vectors derived from differently modified spectrograms. The combination is possible at various stages of TRAP-based probabilistic feature extraction. These combination schemes are suggested and examined.

Chapter 7 examines the performance of ASR with TRAP-based probabilistic features on noisy speech. First, the task is defined and ASR system is described. Then, TRAP vectors with different kind of normalization are tested. Further we proceed with normalisation which performs the best.

[3]Standard features exists for several decades and the rest of ASR is tuned to them. The novel features thus have much harder way to go.

The feature extraction schemes which gave good results in previous three chapters are tested on this task to compare their robustness in noisy conditions.

Chapter 8 shows the performance of the features in the state-of-the-art large vocabulary continuous speech recognition systems. The quality of the features is evaluated on two tasks: recognition of meetings recordings and conversation telephone speech recognition.

For meeting speech recognition, the choice of optimal TRAP-based probabilistic features was based on the results from previous tasks. In this optimization step, hybrid [10] approach was used, giving us an advantage of fast system evaluation. Then, these features were used in GMM-HMM system. Also, combination of TRAP-based probabilistic features with standard MFCC features was examined.

The conversation telephone speech task was defined at ICSI, Berkeley. Here, TRAP-based features are part of more complex feature extraction. The techniques examined in previous chapters were used to improve the system.

Chapter 9 gives an overview of current work of other researchers relevant to TRAP-based probabilistic feature extraction:

- extraction of long-temporal narrow-frequency information from input speech (temporal pattern extraction)
- parametric representation of temporal patterns
- structure of temporal classifier

Further we introduce Universal TRAP and split context technique and extraction of spectro-temporal patterns.

Then, the work is summarized and the future plans are outlined.

1.4 Claims of this thesis

The original contributions of this thesis can be summarized as follows:

- **Step by step development of TRAP features and their evaluation on small vocabulary digit recognition task**. This can serve as a guide to TRAP based techniques and will help to find a place of further research.
- **Test of the promising techniques on noisy speech**. This provides insight on behavior of different TRAP-based features in noisy environment. Our techniques can be used as baseline for development of noise robust speech recognition.
- **Combination of TRAP-based probabilistic features with standard cepstral features in state-of-the-art large vocabulary continuous speech recognition systems**. Here, the usefulness of probabilistic features for recognition system is shown. The complementarity of TRAP-based probabilistic features and cepstral features brings significant improvement in system performance when both features are used for GMM modelling. These results should guide other researchers in building accurate and reliable LVCSR system.
- The importance of **proper processing of TRAP vector** is shown throughout the thesis and well-founded by experimental results over several experimental setups.
- **Pioneering research in area of critical band spectrogram modification**. The technique is used also in Multi-resolution RASTA [35] technique which was partially motivated by our research.

Chapter 2

Bases of TRAP

2.1 Motivation

Standard features (MFCC, PLP) are based on speech magnitude spectrum in one time frame. If the noise occurs in speech signal, the spectrum will be impaired. If the noise is frequency limited (real noises are usually characterized by some frequency band where they occur) only a part of the spectra will be affected. But due to the decorrelation on the end of feature extraction (point 5. in section 1.1.1.1) the impairment will be spread over the entire feature vector.

Multi-band approach [78] tries to solve this problem by running several speech recognizers in different frequency bands and recombining their results. Features for each stream are computed from given frequency band only. We can split the spectrum into several streams to achieve greater noise robustness of the recognition system (see Fig 2.1). Further increase of number of streams will reduce the number of critical-band spectral points the features are computed from. Finally, we would have a recognizer for each point of critical band spectrum. But there is only minimum information in one point of spectrum and such system would have poor performance. We can increase the amount of information in streams by adding time context. Thus we get the temporal trajectory in one frequency band instead of frequency vector in one time frame. Fig. 2.2 presents a logarithmic critical band spectrogram where the frequency vector, used for standard feature extraction, and temporal pattern, used in our approach, are shown. The crossing point of the two vectors is marked by the star. It points the actual time and frequency the vectors are derived for.

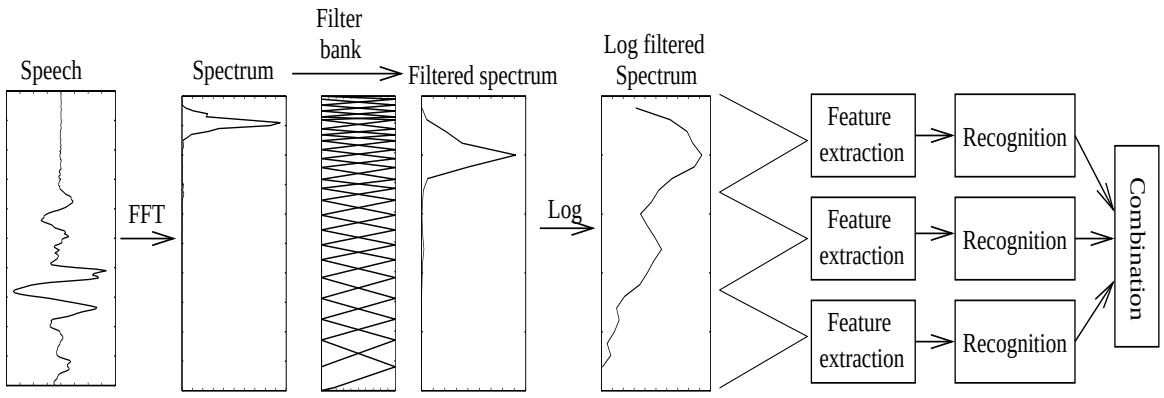


Figure 2.1: Block diagram of Multi-band system

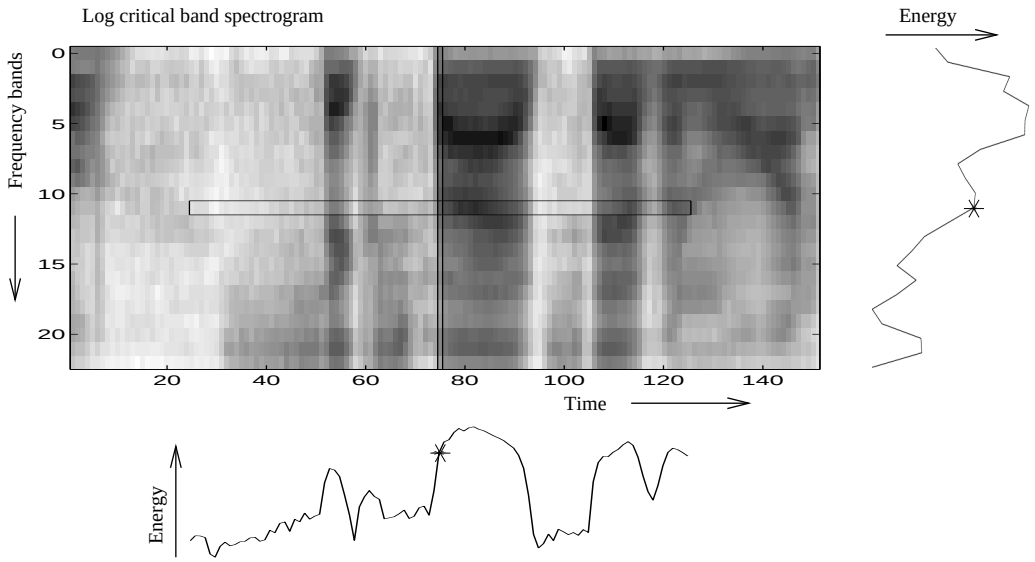


Figure 2.2: Logarithmic critical band spectrogram with frequency and time vector

2.1.1 Narrow-band frequency

We came to narrow band representation from the contemplated noise-robust multi-stream system. But this idea was already drawn by H. Fletcher in his human listening experiments [25]. Fletcher has hypothesised that independent, narrow-band frequency detectors, working in parallel, account for the robustness of human auditory processing. This hypothesis was further supported by other listening experiments: Greenberg, et. al. [30, 81] and Warren, et. al. [93, 92] independently show how human can still well recognize words even if only several narrow frequency bands are kept and the rest is filtered out.

Another set of experiments examined identification of consonants in narrow-band signal [61]. G. Miller and P. Nicely observed, that the errors are not random but follow some patterns. They found that errors are more often made in consonants sharing an attribute like voicing, nasality, affrication and place of articulation. Consonants not sharing an attribute are only rarely confused.

The frequency bands used in our work have the "critical" bandwidth. This term comes from early Fletcher's listening experiments where he studied the perception of pure sinusoidal tone in noise. The noise was band limited and centered at the frequency of sinusoidal tone. When the frequency band of noise was widened, the energy of sinusoidal tone had to be increased in order to be perceived by the listener. But this happened only within certain bandwidth – further increase of noise bandwidth did not effect perception of sinusoidal tone. The maximum bandwidth of the noise, which affects the perception of the sinusoidal tone at its central frequency is called "critical band".

2.1.2 Long time context

The use of longer context is supported by psychoacoustic studies. J. A. Bilmes and H. H. Yang et. al. have shwn in [8, 96] that the information about current phoneme is spread over neighboring phonemes due to the coarticulation – fluent transition of speech production organs from one configuration to another. The phonemes are not completely acoustically separated, they overlap. The information theoretic analysis has shown that significant discriminant information about the current phoneme is at times up to several hundreds milliseconds away from it. This suggest that to have complete information about a phoneme, a segment of a length of half second is needed.

Another support for longer time context came from studies of modulation frequencies important for

speech recognition [50]. It was shown that most important frequencies are between 2 and 16 Hz with maximum at 4 Hz. 4 Hz frequency corresponds to time period of 250 ms, but to capture frequencies of 2 Hz, the interval of half second is needed.

In standard features, the temporal information is represented by delta features. They usually span time interval of about 100 ms. Such interval is very short compared to the values given above. In our approach, we would like to preserve all the information which can be used for phoneme classification. That is why we chose time interval of one second. The center of this interval corresponds to the actual time.

2.1.3 Probabilistic features

Why are the classes probabilities supposed to be good features for automatic speech recognition? Ideally, we would like such features, that have maximal mutual information between the feature vector $\mathbf{x}$ and the class Q_i they belong to. It has been shown, that maximizing the *aposteriori* probability of class maximizes also the mutual information $I(\mathbf{x}, Q_i)$, under the condition that all classes Q are equally likely [10]. The classes above are denoted as in Eqs. 1.2 to 1.5 to emphasise the importance of consistency between classes used in probabilistic feature extraction and classes used in subsequent acoustic model.

An ideal feature extraction should be able to reduce the error to its theoretical limit, which is given by Bayes' error [27]. For K class problem, the Bayes classifier compares aposteriori probabilities of vector $\mathbf{x}$: $p(\mathbf{x}|Q_i)$ for all classes and classifies $\mathbf{x}$ to the class with maximum aposteriori probability. Since aposteriori probabilities are not linearly independent, as

$$\sum_{i=1}^{K} p(\mathbf{x}|Q_i) = 1, \tag{2.1}$$

only $K-1$ probabilistic features would be the ideal set of features which would give the Bayes' error.

To estimate class aposteriori probabilities, the discriminative connectionist model – artificial neural network (ANN) – is used. This model learns the transform of the input vector $\mathbf{x}$ to aposteriori probability directly from the data.

The discriminative training of the model focuses on the boundary between the classes where the differences are magnified, whereas the details in the "middle" of the class are rather minimized. This transformation makes the resulting probabilistic features more separable. This issue was discussed in [34].

The probabilistic features offer one more advantage over conventional features: easy combination of different features. This is possible on the level of probabilities level where we can combine them by simple averaging of output probabilities from different ANNs. If the combined streams provide better probability estimates, then – based on above discussion – we also obtain better probabilistic features which should result in improved ASR performance.

2.2 Detailed description of Basic TRAP system

As mentioned above, the key element of our approach is the evolution of energy in a narrow frequency band – critical band. The shape of energy over the time of one second is called "TempoRAl Pattern" – TRAP – vector.

The TRAP vectors are obtained as follows: The speech signal is segmented into 25 ms frames with 15 ms overlap. The spectrum of speech segment is computed by the Fast Fourier Transform (FFT). We take the power of the spectrum and filter it by a bank of critical band filters. Usually Bark scaled trapezoidal filters are used. The logarithm is taken at the output of the filters. Here, log-critical band spectrum (shown in Fig. 2.2) is obtained. We can notice, that the processing so far is the same as

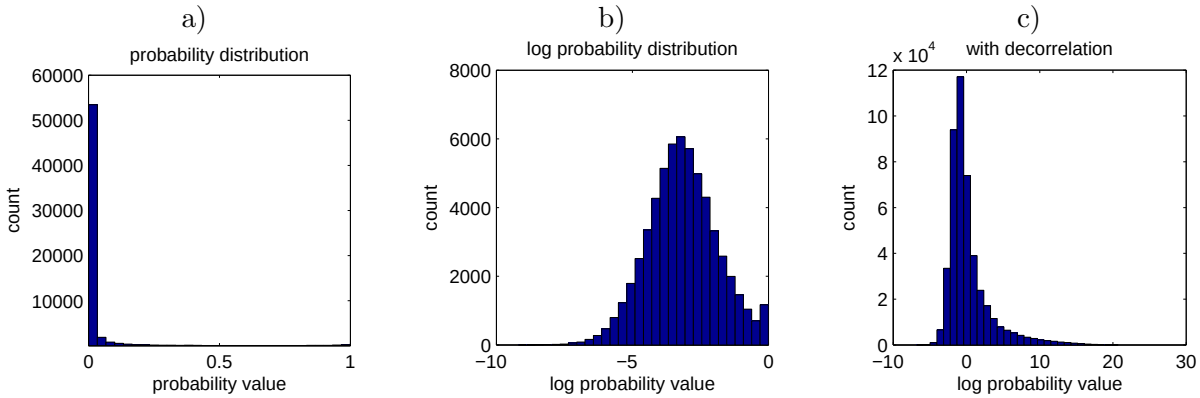


Figure 2.3: Histograms of a) output class probability, b) log of the output probability, c) decorrelated coefficient

for standard features and corresponds to the first half of the processing in Fig. 2.1. Then, each point of the log-critical band spectra (output of one critical band filter) is appended by 50 frames of time context on both sides. Such 101 point TRAP vector covers 1 second of original speech signal.

We can imagine that the extension of one critical band spectrogram point by the context represents the feature extraction in Fig 2.1. The TRAP vector is now feature vector for subsequent band-conditioned recognition (classification). This is done by an Artificial Neural Network (see below) which classifies the TRAP vector to speech class. As speech classes, we choose the phonemes – besides the reasons mentioned in section 2.1.3, also for the fact that there exist phonetically labeled (or aligned) databases and the number of the classes is reasonable. The output of ANN is a vector of phoneme posterior probabilities. Since these probabilities are estimated only from one band, we will refer to them as band-conditioned class probabilities. The ANN is referred to as *band-conditioned classifier* or *band-conditioned probability estimator*.

Then, all band-conditioned class probabilities are concatenated to a single vector. This vector is processed by a negative logarithm to obtain better distribution of the parameters (see below). Then the vector is fed to another ANN which does the combination of the band-conditioned estimates and produces the final probability estimation. This ANN is called *merger* or *merging classifier*. The output classes are usually the same as for band-conditioned classifiers.

The cascade of band-conditioned and merger classifiers is called *temporal classifier*.

The class probabilities are then converted to probabilistic features suitable for the standard GMM-HMM recognizer. The GMM-HMM system has two assumptions about the features which have to be fulfilled:

1. **Gaussian distribution** – The feature distribution is modelled by a mixture of Gaussian functions. The distribution of probabilities lies between zero and one – see Fig 2.3a. If the classification was perfect, the histogram would have only two peaks – at zero and at one. Since our classification is not perfect, it is easier to tell which class is not the one the vector belongs to. There also are more input vectors which do not belong to the given class. For these two reasons, we have a big peak at zero and smaller (if at all) at one.

   Such distribution can not be easily approximated by Gaussians. The log is applied on the ANN output to spread the paek at zero. To avoid problems with infinity the probability is floored[1] The distribution of log probabilities can be seen in Fig. 2.3b. The log has spread the peak which was at zero, but still there is a sharp end which was at one.

[1] A good foor is 10^{-10}.

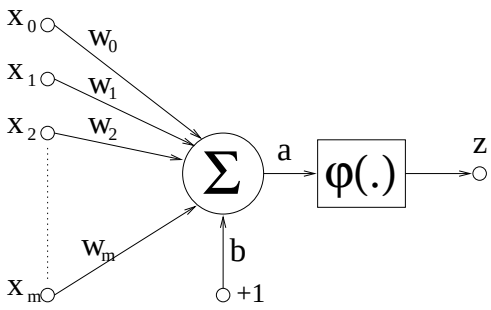


Figure 2.4: Scheme of artificial neuron with input vector $\mathbf{x}$, weights w_i, bias b and activation function $\varphi(\cdot)$.

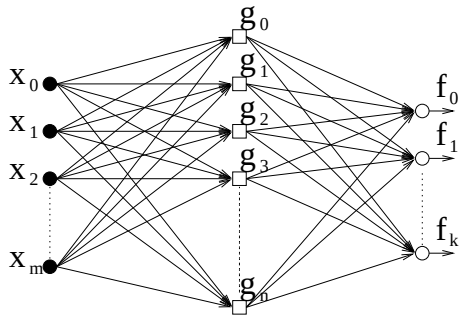


Figure 2.5: Scheme of three-layer artificial neural network

2. **Decorrelation** – The covariance matrix of Gaussians mixture models is usually diagonal, which means that the features have to be decorrelated. The class probabilities are correlated – if a certain class has low probability, then acoustically similar classes have also low probabilities – so we need to add the decorrelation in extraction of probabilistic features. It is done by the Principal Component Analysis (PCA). It can also be used for dimensionality reduction if desired.

At the end of this processing, features suitable for standard HMM-GMM recognition system are obtained. These features are called *TRAP-based probabilistic features.*

2.2.1 Artificial Neural Network

Artificial Neural Network (ANN) is an interconnected group of artificial neurons (nodes). The neural networks are non-linear statistical data modelling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. While a neural network does not have to be adaptive, its practical use comes with algorithms designed to alter the strength (weights) of the connections in the network to produce a desired signal flow. Then the ANN can be used as an arbitrary function approximation mechanism which 'learns' from observed data.

The ANN defines a function $f : X \rightarrow Y$. The word "network" in the ANN term is used because the function $f(\mathbf{x})$ is defined as a composition of other functions $g_i(\mathbf{x})$, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables. A widely used type of composition is the nonlinear weighted sum:

$$f(\mathbf{x}) = K\left(\sum_i w_i g_i(\mathbf{x})\right) \tag{2.2}$$

where $K(\cdot)$ is some predefined function. A scheme of an artificial neuron, which performs function $g_i(x)$ and has an output z is shown in Fig. 2.4. The output of the neuron is

$$z = \varphi\left(b + \sum_{i=0}^{m} w_i x_i\right) \tag{2.3}$$

where b is bias, w_i are weights and $\varphi(.)$ is the non-linear activation (or transfer) function. As activation function, we use the sigmoid (or logistic) function, which is defined as:

$$z = \frac{1}{1 + \mathrm{e}^{-a}} \tag{2.4}$$

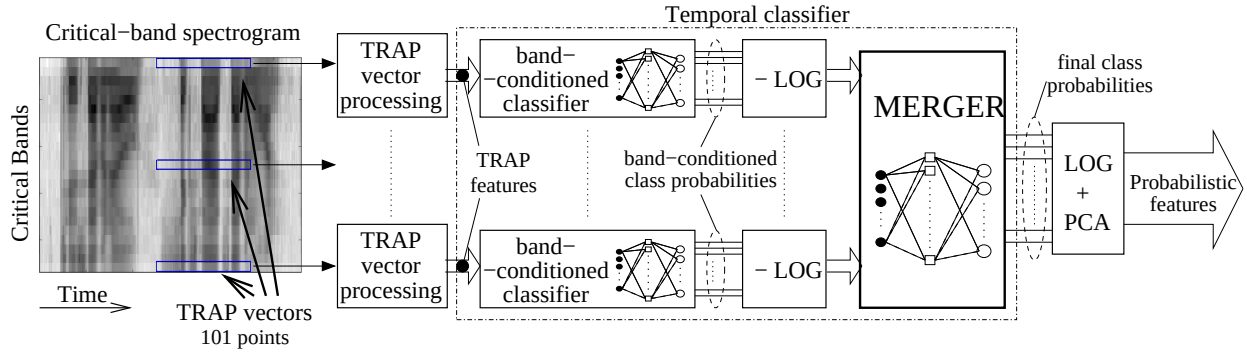


Figure 2.6: Block diagram of basic TRAP system

In our work, we use the simple type of ANN where the information moves in only one direction, forward: from the input nodes, through the hidden nodes, to the output nodes, without any cycles or loops. Such network is called feedforward neural network. We use a three-layer[2] network with fully connected layers – each neuron in one layer has a connections to all neurons in the subsequent layer. Such neural network belongs to the class of Multi-Layer Perceptrons (MLPs). The scheme of the MLP is in Fig. 2.5. The MLP in our work is used as the pattern classificator and the output function f_i estimates the posterior probability of i-th class.

We train the MLP in supervised manner[3] , that is we have a set of input–output example pairs $(\mathbf{x}, \mathbf{y}), \mathbf{x} \in X, \mathbf{y} \in Y$ and the MLP has to learn to model the dependency, function f, between them. We define a cost related to the mismatch between our mapping and the data. It implicitly contains prior knowledge about the problem. A commonly used cost is the mean-squared error

$$C = \sum_n ||\mathbf{f}(\mathbf{x}_n) - \mathbf{y}_n||^2 \tag{2.5}$$

which tries to minimise the average error between the network's output, $f(\mathbf{x})$, and the target value $\mathbf{y}$ over an n example pairs $(\mathbf{x}, \mathbf{y})$.

Since our output values are a probability estimates lying between 0 and 1, an appropriate cost function is so called cross-entropy

$$C = \sum_n \sum_i y_{ni} \ \ln(f_i(x_n)) + (1 - y_{ni}) \ \ln(1 - f_i(x_n)) \tag{2.6}$$

Minimising a cost function using gradient descent leads to the well-known back-propagation algorithm [75]. The training examples are presented repeatably to the MLP until weights converge or the error is acceptable.

2.3 TRAP vector processing

The feature extraction step in Fig 2.1 can be more complex than just adding context to the actual point. The TRAP vector can be processed in different ways to increase accuracy of the following temporal classifier and thus increase quality of the probabilistic features. The block diagram of TRAP based probabilistic feature extraction is shown in Fig. 2.6. The common TRAP vector processing steps are:

[2]Note that the first layer is not always considered a real neural network layer.

[3]Pattern recognition – classification – is one of the tasks suitable for supervised learning.

Normalization: The normalization removes the information about absolute energy level in the critical band and helps to focus on the shape of the TRAP vector. It also reduces the impact of noise on TRAP.

Mean normalization reduces the effect of convolutive noises such as different transmission characteristic of technical equipment used for the speech signal acquisition (microphone's characteristics, telecommunication line's characteristics, ...). Those characteristics are additive in log-spectral domain and therefore can be subtracted.

Variance normalization is useful mostly when additive noise is present. The acoustic noises are additive in time domain and therefore cannot be easily removed from spectral representation. Variance normalization will alter the dynamic range of the vector and thus make the noiseless and noise signals similar.

Hamming windowing: TRAP vector can be weighted by Hamming window to emphasize center part of the vector. We assume that TRAP vectors from the same class will have biggest similarity in the center of the vector. The coarticulation, captured further from the center, helps to classify the current phoneme, but the distant coarticulation is not so important.

Projection of TRAP vector on bases vectors: The multiplication of a TRAP vector by a matrix of bases is done in order to preserve or enhance important properties of TRAP vector. Mostly, the dimensionality reduction is also done in this step. The bases used for projection can be analytic bases such as DCT, or the bases vectors can be obtained from training data using a data analysis techniques such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA), or these bases can be designed ad-hoc.

The first two mentioned TRAP vector processing steps are done in almost all of the presented experiments. The mean and variance normalization of TRAP vector followed by Hamming windowing will be further referred as *basic processing*. To distinguish between the original TRAP vector and the processed one, the output of TRAP vector processing is called TRAP features. These features create input for further classification.

2.4 TRAP system evaluation

To compare two different systems, the evaluation of system performance is needed. In speech recognition, the basic evaluation of the system is word error rate (WER) which indicates how often the system doesn't recognize the word. The lower the WER is, the better the system performance is. The WER is computed by:

$$WER = \left(1 - \frac{hits - insertions}{number\ of\ words}\right) \times 100\%. \tag{2.7}$$

However, for TRAP-based feature extraction we would like to have some evaluation before running the final recognizer, which can be big and can require a lot of computation resources. One of the possible measurements are the frame classification accuracies recorded during the neural net training (see section 3.4.1). We are mainly interested in merger classification accuracies. The overall classification accuracies are good for basic comparison of classifiers and processing done on TRAP vectors, but the trends in these accuracies often do not correspond to trends in WERs.

Chapter 3

Stories – Numbers: experimental setup

Stories – Numbers experiments were conducted exactly on the same experimental setup and data sets S. Sharma [78], P. Jain [45] and J. Černocký [90] used for their work. Therefore the results are directly comparable. Besides the advantage of getting comparable results with others researchers, this experiment was defined mainly because it was small and fast enough to test various TRAP-based probabilistic features. The aim of these experiments was to test various TRAP-based probabilistic features focusing on the extraction of TRAP features for subsequent classification. Neither the question of target classes nor the conversion of probabilities to probabilistic features are addressed here.

In this chapter the experimental setup of the Stories – Numbers experiment is described together with software used for ANN and GMM-HMM training. The ANN training process is also valid for other tasks presented in this work.

3.1 Task

The task is to recognize continuously spoken digits "zero" to "nine" and "oh" from the telephone quality speech. Testing recordings contain one to seven digits obtained from queries like "apartment number", "ZIP code" or "telephone number". No prior knowledge is used in this task, so there is no constraint on the maximum number of digits in recognized sequence. The only condition is, that at least one digit has to be recognized.

3.2 Used data

For this experiment, two databases were used. The first one is **OGI-stories** [16] database. It was used for training of the band-conditioned classifiers. This database contains 688 speech files of which 208 have correct phonetic transcription. These files were used for training. They contain 989 518 frames which corresponds to 2 hours and 44 minutes of speech.

The second database is **OGI-numbers** [73]. It consists of 15000 speech files of which 6640 have phonetic transcription. Two sets are created from this labeled data: The first one contains not only digits, but also numbers. This part was used for training of merger classifier. It contains 3590 sentences with 603 377 frames which corresponds to 1 hour and 52 minutes of speech.

The second set consists of files which contain only digits. It is used for training of the GMM-HMM recognizer. This set has 2547 sentences with 458 582 frames which corresponds to 1 hour and 16 minutes of speech. This set is subset of the first one used for merger training.

The test set is also derived from OGI-numbers database. It contains files with digits only without phonetic transcription. There are 2169 sentences (12437 words) with 626 589 frames which corresponds to 1 hour and 44 minutes of speech.

The overview of used database parts is given in Tab. 3.1.

Used for	band-conditioned ANN train	merger ANN train	GMM-HMM train	test
source	OGI-stories	OGI-numbers	OGI-numbers	OGI-numbers
labeled	YES	YES	YES	NO
time	2h 44m	1h 52m	1h 16m	1h 44m
frames	989 518	675 177	458 582	626 589

Table 3.1: Data sets used in Stories – Numbers experiments

3.2.1 Analysis of test data

The purpose of this analysis is to find out whether the 100% recognition accuracy is possible. The recordings were listened and compared with the reference transcription in search for obvious problems for given task, which is mainly disagreement in what is actually spoken and the transcription. If such files exist, the recognition accuracy cannot be 100% and the system can reach only a certain accuracy level.

To avoid checking all files, the results from one of our experiments were used. The test utterances (files) which had word recognition accuracy lower then 35% in our baseline `base_trap101_mvn` (section 4.1) were checked. Thus we obtained the information about utterances which are potentially problematic. The summary of the analysis is given in Tab 3.2.

The utterances not suitable for this task were detected. These utterances contain other speech than the numbers. The recognizer used in our experiments is forced to output digits whenever there is a speech in the input file, but such output will be always wrong since the transcription of such file contains only the digits and the other speech is ignored. Recognition of these utterances cannot be improved without increasing of complexity of recognition system.

inserted speech	5
strong noise in speech	1
other noise in speech	8
unintelligible (completely or partly) utterances	5
intelligible utterances without obvious problem	10

Table 3.2: Summary of analysis of the 29 least accurate utterances

3.3 Critical band parametrization

The speech files were converted into logarithmic critical band energies using `rasta` executable from `ICSI Sprach Core` package [22]. We executed the `rasta` code with following command-line parameters: `-M -A -R -L -w 25 -s 10`. Depending on input file format the parameter for input format is also needed. This command will:

- divide speech signal into 25 ms frames with 10 ms shift,
- compute power spectrum of framed signal,
- filter it by 15 Bark scaled trapezoidal filters,
- take the logarithm,
- store the output for each utterance in ASCII file, one line per frame.

Due to the time context of TRAP vector, it is necessary to add artificial context on both sides of the file. Since we are using 101 point long TRAP vector, the appended context will be 50 frames. We obtain these frames by flipping the ends of file. Thus we are sure, that there is no discontinuity and that the values are natural. Note that the number of frames will increase by 100 per one utterance.

3.4 Multi-layer perceptron

The Multi-Layer Perceptron (MLP) is a very neutral, generic and the most widely used type of artificial neural networks – general connectionist models. It is both simple and based on solid mathematical grounds. It has been shown, that MLP with one hidden layer can approximate any nonlinear function with arbitrary accuracy, given enough neurons in hidden layer and enough training data points [41].

The bases of Artificial Neural Network were introduced in Sec 2.2.1. In this section we give more detailed information about used MLP and the way it is trained in the QuickNet toolkit – software handling MLPs.

This software is also part of the `ICSI Sprach Core` package [22]. It allows fast and effective training of a simple, fully-connected, feed-forward MLP with a single hidden layer. The software was optimized for the pattern classification task specially in speech recognition. It supports several types of MLP non-linearity as well as different training criteria. The following sections will describe settings used throughout all the work.

3.4.1 Neural net training – learning rate

Training the network weights is performed with a large training set of example input/output pairs. The goal of the training procedure is that the neural net generalizes across these relationships in its internal representations, so that previously unseen input results in an appropriate output.

The neural net training procedures are based on the original back-propagation algorithm [94], in which the partial derivative of an error measure with respect to each weight in the network is used to shift that weight a little bit in the opposite direction (i.e. to reduce the error). The size of this shift is governed by a parameter called the learning rate, which is a compromise between speed of learning and precision of result.

To achieve the generalisation, an early-stopping criterion is used to determine the end of the training. In case the neural network is too complex for given task, i.e. it has too many free parameters for given training set, it would learn perfectly the training data, but the generalisation would be poor. To prevent such situation, part of the training data – cross-validation set (CV) – is held aside and the MLP performance is evaluated on it throughout the training. When the MLP starts to over-fit the training data, the performance on the cross-validation set drops and the training procedure is terminated.

The training procedure makes multiple passes through the entire set of training patterns; each pass is called an epoch. After each epoch, the performance of the MLP is evaluated in terms of frame accuracy (FA) using the cross-validation set which size is usually about 10% of all training data. The cross-validation frame accuracy (CVFA) is the portion of correctly classified patterns in this set – the MLP output unit generating the highest value matches the class of the input pattern.

The learning rate scheduler, known as “newbob”, starts with a reasonably fast learning rate of 0.008. This learning rate is kept for subsequent epochs until the increment in CVFA is less than 0.5% over the previous epoch. After that, the learning rate is halved for following epochs. This leads to increasing precision ending up in the local optimum. Initially, reducing the learning rate leads to a big boost in CVFA, but eventually the learning rate becomes so small that the improvements are minimal. Training ceases after an epoch in which the CVFA again improves by less than 0.5%.

3.4.2 Neural net training – output non-linearity

Usually, the MLP does not have any non-linearity in output layer and the output is only a weighted sum of hidden layer outputs. We want the outputs of an MLP to be interpretable as posterior probabilities for a categorical target variable. It is thus desirable for those outputs to lie between zero and one and to sum up to one. The purpose of the *softmax* activation function is to enforce these constraints on the outputs. Let the input to activation function of output node be $q_i, i = 1, \ldots, c$, where c is the number of output nodes (also categories). Then the softmax output p_i is:

$$p_i = \frac{e^{q_i}}{\sum_{j=1}^{c} e^{q_j}} \tag{3.1}$$

With such output non-linearity, the output layer activations are directly interpretable as posterior probabilities of a set of mutually-exclusive classes.

3.4.3 Further details of neural net training

Minimum cross-entropy error criterion (Eq. 2.6) is used.

The error is back-propagated after each input pattern. This requires more computation during MLP training, but the training algorithm converges faster. The back propagation after a bunch of training vectors would be advantageous if more data were available. Then the time saving would be considerable and the convergence would be still fast because of more data.

Before the training takes place, mean and variance normalization is done over all training data, in order to get the input data to certain range. In this way, fast progress in training is ensured because the training algorithm does not have to look for data distribution. The unseen test data are normalized by stored means and variances estimated on the training data.

The nets are trained against the hard target labels. This means that only one value from target vector is nonzero and equal to one.

3.5 Phoneme classes

As mentioned above, the training data is labeled. The labels provided with the databases cannot be used directly for two reasons: First, the time boundaries are given on the sample level. This was solved by rounding the times to tens of milliseconds which corresponds to frame rate. Second, the labels contain details about the manner of pronunciation. If all labels were used, the number of classes would be high and the MLP would be forced to distinguish between very similar events. Also, some classes with very low number of occurrences were found. To obtain reasonable number of target classes with sufficient number of training vectors, the following steps were done:

- All information distinguishing centralized, lengthened, more or less rounded, etc. phonemes was removed.
- All non speech sound events such as breath and lip noise, cough, sneeze, etc. were labeled as silence.
- Phonemes with insufficient number of occurrences were mapped to acoustically similar classes. These phonemes are `ix`, which was mapped to phoneme `ih`, and `ng`, which was mapped to phoneme `n`.

Having both databases labeled in the same way, the phoneme classes common to both of them were chosen to be a target classes for both band-conditioned and merger classifiers. The chosen set contains 29 phonemes, and is determined by the phonemes in OGI-numbers. They are (in OGIbet

label	index	Stories		Numbers	
		count	perc%	count	perc%
d	0	5687	0.70	441	0.06
t	1	19606	2.42	22200	3.28
k	2	12956	1.60	2765	0.40
dcl	3	13761	1.70	521	0.07
tcl	4	27241	3.36	21215	3.14
kcl	5	17250	2.13	6800	1.00
s	6	50978	6.29	39880	5.90
z	7	14635	1.81	7003	1.03
f	8	17131	2.11	28389	4.20
th	9	5162	0.64	11728	1.73
v	10	9675	1.19	14977	2.21
m	11	20948	2.59	38	0.00
n	12	36695	4.53	50424	7.46
l	13	23079	2.85	916	0.13
r	14	20860	2.57	29717	4.40
w	15	15043	1.86	20485	3.03
iy	16	34554	4.26	32362	4.79
ih	17	38111	4.70	16640	2.46
eh	18	22200	2.74	13970	2.06
ey	19	20328	2.51	19085	2.82
ae	20	26146	3.23	162	0.02
ay	21	28054	3.46	53922	7.98
ah	22	49583	6.12	28219	4.17
ao	23	5867	0.72	4027	0.59
ow	24	16639	2.05	47529	7.03
uw	25	11086	1.37	28103	4.16
er	26	15137	1.87	2633	0.38
ax	27	11301	1.39	853	0.12
sil	28	220575	27.22	170173	25.20
total		810288		675177	

Table 3.3: Phoneme coverage in Stories and Numbers

notation – see Appendix A):
`d t k dcl tcl kcl s z f th v m n l r w iy ih eh ey ae ay ah ao ow uw er ax sil`

The frames from OGI-stories with labels different from the target ones were not used for training. But the segments belonging to these non-target phonemes are not stripped out of the training set – they serve as natural context of the target phonemes enriching the context variation and thus also increasing the robustness of the band conditioned classifiers.

The coverage of target classes in OGI-stories and OGI-numbers databases is given in Table 3.3.

3.6 HMM recognizer

The HMM-GMM recognizer is built using the hidden Markov model toolkit (HTK) [97]. The recognizer uses context independent phoneme acoustic models. Only 23 phonemes occurring in digits are modelled: `w ah n ow th r iy f s eh v ih tcl t uw kcl ay ey k ax ao z sil`

Each phoneme model consists of five emitting states. The emission probability distribution is modeled by three Gaussian components with diagonal covariance matrices. The recognizer dictionary contains digits "zero" – "nine" and "oh", with some multiple pronunciations. Together, there are 29 pronunciation variants. This simple recognizer does not have any language model, as all words are equally likely and can occur with the same probability independently on the previous words.

The steps of training of acoustic models are the following:

1. initialization of models using phonetic transcriptions by `HInit`.
2. 3 iterations of basic Baum-Welch re-estimation of the models by `HRest`.
3. 5 iterations of embedded Baum-Welch re-estimation of the models by `HERest`.

The decoding was performed using `HVite` tool with cross-word transition log penalty of -25.5 (this number was found optimal by S. Sharma [78]). The scoring was done using standard tool `HResults`.

The performance of such a simple recognizer with standard MFCC features consisting of 12 cepstral coefficients appended with log energy plus their delta (velocity Δ) and double-delta (acceleration $\Delta\Delta$) coefficients is **5.9%** WER.

3.6.1 Significance test

When the experiments are carried out, we would like to know, if the results are statistically different – i.e. if such results cannot occur by chance. The statistical test of significance is a way to find it.

Generally, statistical tests assume that outcomes of experiments are samples of some populations. If the differences in results are large, it is less likely that they came from the same population. Then we can say the results are different (i.e. came from different populations) with certain level of confidence, or that there is certain (small) amount of probability that the results came from the same process, i.e. are the same. The amount of probability that the processes are the same is called significance level α. The smaller the significance level is, the stronger has to be the evidence to say that the processes behind the results are different.

In case of word recognition, where each word can be assumed as independent yes/no experiment, the samples are taken from binomial distribution $B(n, p)$ where n is the number of independent trials (words) and p is the probability of incorrect answer[1]. The number of incorrectly recognized words in the test set is y. The distribution mean is $\mu = np$ and variance is $\sigma^2 = np(p-1)$.

When two experiments are performed, they are samples from a binomial distribution $B(p, n)$. The respective numbers of incorrectly recognized words in test sets are y_1 and y_2. We assume a null hypothesis **H0** under which the two results are obtained from the same distribution, i.e. have the same "true" mean and a alternative hypothesis **H1** under which the results came from different distributions and thus have different means. Since we want to test whether one process is better than another, the hypothesis are:

H0 : $p_1 = p_2$ — the scores are really identical

H1 : $p_2 < p_1$ — p_2 is indeed better than p_1 at the current significance level α.

Considering the relative difference between the two results $d = (y_1 - y_2)/n$, then under **H0** hypothesis the mean and variance of such difference d are:

$$\mu_d = \mu\left(\frac{y_1 - y_2}{n}\right) = p_1 - p_2 = 0 \tag{3.2}$$

$$\sigma_d^2 = \sigma^2\left(\frac{y_1}{n}\right) + \sigma^2\left(\frac{y_2}{n}\right) = \frac{p_1(1-p_1)}{n} + \frac{p_2(1-p_2)}{n} \tag{3.3}$$

[1]The incorrect answer is there to be consistent with mostly used measure in ASR – word error rate – the probability of incorrect recognition.

Note that even if we consider the same underlying process for both results, so that the true difference mean is zero, the variance of the difference is taken as if the two processes were actually different. The resulting distribution of d is approximately normal with n large enough. Now we want to know how likely it is to obtain the same or bigger difference than the one observed, assuming the above distribution. Such probability can be computed from cumulative distribution function which, for normal distribution, is[2]:

$$F(x;\mu,\sigma) = \mathrm{P}(X \leq x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{(u-\mu)^2}{2\sigma^2}\right) du \tag{3.4}$$

To avoid integral computation, the difference d can be expressed in terms of standard deviations which in fact transform the distribution to the normal distribution with zero mean and unity variance. Such value is called z-score (or standard or normal score)

$$z = \frac{d - \mu_d}{\sigma_d} = \frac{p_1 - p_2}{\sqrt{\left(p_1(1-p_1) + p_2(1-p_2)\right)/n}} \tag{3.5}$$

This value is then compared with the tabulated values of cumulative distribution functions for given significance level α.

The $\alpha = 0.05$ is usually considered as a border-line for acceptable error level. This level will be used to conclude, if the difference d in results is statistically significant, to say, that the results are different, i.e. one of them is better than another. The z-score value for significance level $\alpha = 0.05$ is $z_{0.05} = 1.645$. Stronger significance for $\alpha = 0.01$ is also considered, with $z_{0.01} = 2.326$.

The result from the basic TRAP experiment in Sec. 4.1 with the probability of incorrect recognition $p_1 = 0.066$ and number of independent trials – words – $n = 12437$ is taken as baseline. To achieve $z_{0.05}$, the second result has to have the probability of incorrect recognition $p_2 \geq 0.0609$ and $p_2 \geq 0.0588$ for significance level $\alpha = 0.01$. In other words, the result has to be better by at least 0.6% WER to consider it as improvement with 95% confidence. To be 99% confident that the second result is really better than our baseline, the improvement has to be at least 0.8% in WER.

The presented approach of significance test can be used only for a simple recognition system. It assumes independence of individual test trials which is not fulfilled when language model is used. Significance test for more complex recognition systems are still possible, but most of them have to be re-evaluated for every experiment [29].

[2] the integral in the equation goes from $-\infty$ so the difference should be taken negative: $x = -d$

Chapter 4

Trap's derived from critical-band spectrogram

Two kinds of TRAP vector processing are tested in this chapter: dimensionality reduction of the vector and merging the vectors from neighboring bands to form the input to band conditioned classifier. All experiments were done in order to find basic tendencies in the final recognition WER. The goal was to improve the system as whole, not to achieve the best possible WER. That is the reason why we keep all the time the length of TRAP vector 101 points and do the mean and variance normalization on it.

Each experiment will be also denoted by its "work name" for further references. The structure of the probability estimators will be given in following form: number of input units – hidden units – output units. In the brackets, there is the number of neural net weights for given probability estimator. The FAs for band-conditioned estimators are given for the estimator from the 5^{th} band (counting from 0), where we observe most speech activity. Usually, the estimator working in this band has the highest frame accuracy from all band-conditioned probability estimators. The accuracy of band-conditioned estimators is dropping towards the ends of the spectrum.

4.1 TRAP baseline – `base_trap101_mvn`

This experiment was done as the baseline experiment. The following experiments will be compared to it. It was also done in order to reproduce the results of P. Jain [45] and H. Černocký [90] under current conditions — Intel CPU architecture instead of SUN SPERTboards; neural net train cache size of 12000 vectors (this variable has big impact on the training, also there were limits due to the memory on SPERTboards); enhanced C-code and shell scripts.

Only the basic processing (mean and variance normalization and Hamming windowing) is done on TRAP vectors to obtain TRAP features.

Band-conditioned probability estimators structure: 101–300–29 (39000 weights)
Merger probability estimator structure: 435–300–29 (139200 weights)

5^{th} band		merger		WER [%]
train FA [%]	CVFA [%]	train FA [%]	CVFA [%]	
43.0	40.0	84.5	81.5	6.6

Table 4.1: TRAP baseline results

4.2 TRAP processing – dimensionality reduction

The possibility of dimensionality reduction of TRAP features using different techniques was examined in these experiments. The main goal was to reduce the dimensionality of classifier input without degradation of system recognition performance. The dimensionality reduction processing step takes place after the basic TRAP vector processing.

The dimensionality reduction is done by projection of TRAP features to a set of base vectors. If the base vectors are in the rows of a matrix $\mathbf{B}$ then the projected vector $\mathbf{a}_p$ is given by a product of the bases matrix $\mathbf{B}$ and input vector $\mathbf{a}$:

$$\mathbf{a}_p = \mathbf{B} \times \mathbf{a} \tag{4.1}$$

The dimensionality of output vector $\mathbf{a}_p$ is given by a number of bases (rows) in matrix $\mathbf{B}$.

The bases obtained using three standard techniques – Discrete Cosine Transform (DCT), Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) – are tested

4.2.1 Discrete Cosine Transform bases

The DCT is the simplest technique and does not require any additional computation in terms of training base vectors. These bases are the same for any data. The DCT is often used in signal processing, especially for lossy data compression, because it has a strong "energy compaction" property: most of the signal information tends to be concentrated in a few low-frequency components of the DCT. This property is used in our experiments although the processed signal is not an ordinary signal as we are working with spectral energies in certain frequency band. Newer the less, we assume similar behavior of critical band energy to common signals due to the naturally limited speed of speech production. Also, computation of spectral energy from overlapping frames eliminates fast changes in the energy trajectory.

According to our studies and tests, the sufficient number of bases is fifty for a 101 point long input TRAP vector. The bases are formed according to:

$$DCT_j(k) = \cos\left(\frac{\pi}{n} j\,(k - 0.5)\right) \tag{4.2}$$

where j is the number of basis vector, $j = 1, 2, \ldots, 50$, n is the length of the vector and $k = 1, 2, \ldots, n$ is index of coefficient in a given base vector. Note that we do not use the 0^{th} DCT basis, DC component. Resulting features obtained by DCT dimensionality reduction of (processed) TRAP vector will be further referred as TRAP-DCT features.

4.2.2 Principal Component Analysis bases

The PCA (or Karhunen-Loève Transform – KLT) is an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance lies on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. The dimensionality reduction can by done by retaining only those coordinates which contribute the most to the variance of data. We assume, that the most important properties of the data are encoded in the high variance principal components.

The principal components are given by the eigen vectors of a covariance matrix which is computed from training data according to:

$$\mathbf{C} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \tag{4.3}$$

where $\mathbf{C}$ is the covariance matrix derived from N input vectors available for training, $\mathbf{x}_i$ is the i-th training input vector and $\mathbf{m}$ is the estimated mean vector:

$$\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{4.4}$$

The eigen value associated with one eigen vector represents the amount of variability preserved by the projection of input vectors to this particular eigen vector. Therefore only several eigen vectors corresponding to the highest eigen values are used as bases for PCA-based dimension reduction. The number of bases was chosen to be the same as for previous DCT bases. Our experiments have also shown that this number of bases is sufficient. The features obtained from (processed) TRAP vector by projecting them on the PCA bases will be further called TRAP-PCA features.

4.2.3 Linear Discriminant Analysis bases

The LDA proposed by Hunt [42] is a data driven technique used to find such linear transformation which best separates classes of the data. Because the transformation aims to separate classes, the assignment of data vectors to classes has to be known. Resulting matrix contains bases sorted by their importance to class discriminability. The dimensionality reduction can be done by projecting the input vector to several most important bases which will preserve most of the information needed for class separation. LDA, like PCA, ensures decorrelation of transformed data. Moreover, the decorrelation is ensured also in each particular class.

Base vectors of LDA transforms are given by the eigen vectors of a matrix $\mathbf{AC} \times \mathbf{WC}^{-1}$. The within-class covariance matrix $\mathbf{WC}$ represents unwanted variability in data and is computed as the average of the covariance matrices of all classes.

$$\mathbf{WC} = \frac{1}{L}\sum_{j=1}^{L}\mathbf{C}_j \tag{4.5}$$

where L is the number of classes and $\mathbf{C}_j$ is the covariance matrix for j-th class.

$$\mathbf{C}_j = \frac{1}{N_j}\sum_{i=1}^{N_j}(\mathbf{x}_i^j - \mathbf{m}_j)(\mathbf{x}_i^j - \mathbf{m}_j)^T \tag{4.6}$$

where N_j is the number of input vectors available for training which belong to class j. $\mathbf{x}_i^j$ is the i-th training input vector belonging to j-th class and $\mathbf{m}_j$ is estimated mean vector for class j:

$$\mathbf{m}_j = \frac{1}{N_j}\sum_{i=1}^{N_j}\mathbf{x}_i^j \tag{4.7}$$

The across-class covariance matrix $\mathbf{AC}$ represents the wanted variability in data caused by differences in classes and is estimated as a covariance matrices of weighted mean vectors of all classes:

$$\mathbf{AC} = \frac{1}{N}\sum_{j=1}^{F}N_j(\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^T \tag{4.8}$$

where N is the total number of vectors available for training and $\mathbf{m}$ is the global mean computed according to Eq. 4.4. Note, that LDA assumes the same normal distribution of data for all classes with full covariance matrix $\mathbf{WC}$.

An eigen value associated with one eigen vector represents the amount of variability necessary for the discriminability preserved by the projection of input vectors to this particular eigen vector. Only

experiment	5^{th} band		merger		WER [%]
	train FA	CVFA	train FA	CVFA	
base_trap101_DCT50_mvn	42.6	40.2	83.8	81.3	6.1
base_trap101_PCA50_mvn	43.3	41.2	84.1	81.6	6.1
base_trap101_LDA15_mvn	40.4	38.2	84.0	80.6	7.3

Table 4.2: Results for TRAP systems with dimensionality reduction

several eigen vectors corresponding to the highest eigen values can be used as LDA transform for the purpose of a dimension reduction. Features derived from (processed) TRAP vectors using LDA dimensionality reduction will be further called as TRAP-LDA features.

For LDA computation, the phonetically balanced database TIMIT [28] was used. The context dependent phoneme models were trained using standard MFCC coefficients and tied-up into 512 states. The database was automatically aligned to these states which served as classes for LDA. An input vectors for the TRAP analysis were formed obtained as follows:

- signal is segmented into 25 ms frames with 10 ms shift
- power spectrum is computed using FFT and integrated into 23 triangular MEL scaled critical bands
- the logarithm is applied
- actual point in time with 50 frames symmetrical context is taken

The LDA bases were computed for each critical band. Only the first 15 bases were kept for dimensionality reduction.

4.2.4 Experimental results

TRAP projected on DCT bases – `base_trap101_DCT50_mvn`

50 cosine bases were used for projecting of TRAP features.

Band-conditioned probability estimators structure: 50–300–29 (23700 weights)
Merger probability estimator structure: 435–300–29 (139200 weights)

TRAP projected on PCA bases – `base_trap101_PCA50_mvn`

50 PCA bases were used. The bases were computed for each band on the band-conditioned probability estimator training data (labeled OGI-stories database) and stored. For the other data sets, these precomputed bases were used. Note that the database used for training of band-conditioned probability estimator is different from other data sets. This difference may cause suboptimal bases for those data sets and consequently degradation in final accuracy.

Band-conditioned probability estimators structure: 50–300–29 (23700 weights)
Merger probability estimator net structure: 435–300–29 (139200 weights)

TRAP projected on LDA bases – `base_trap101_LDA15_mvn`

15 precomputed LDA bases (see Sec. 4.2.3) were used.

Band-conditioned probability estimators structure: 15–300–29 (13200 weights)
Merger probability estimator structure: 435–300–29 (139200 weights)

The results are shown in Tab. 4.2. It can be seen than the frame accuracies for DCT and PCA dimensionality reduction are very similar to the TRAP baseline (Tab. 4.1). A slight improvement was gained in WER. The results for LDA dimensionality are slightly worse in all cases.

The results have shown that dimensionality reduction is possible without significant degradation of the system performance. Moreover, slight improvement was achieved when DCT or PCA bases are used.

experiment	5^{th} band		merger		WER [%]
	train FA	CVFA	train FA	CVFA	
2b_trap101_mvn	47.4	44.3	85.8	81.7	5.7
3b_trap101_mvn	50.7	45.7	85.7	81.7	5.5

Table 4.3: Results for systems with band merging

4.3 Band merging

Band merging is a technique, which concatenates TRAP features from neighboring critical bands on the input of the band-conditioned probability estimator. Thus, more information is present on the input of the classifier and more accurate classification is possible. This should help merger classifier to make better final estimates and subsequently lower the WER of the system.

Two concatenating schemes are investigated first:

- Concatenate TRAP vectors with basic processing (mean and variance normalization followed by Hamming windowing) from two neighboring bands. Inputs for neighbouring band-conditioned estimators overlap by one critical band. Thus, if the number of critical bands is N, the number of band-conditioned estimators is $N-1$. In our case there are 15 critical bands and the number of band-conditioned band probability estimators will be 14.

- Concatenate TRAP features (TRAP vectors with basic processing) from three neighboring bands. Inputs for neighbouring band-conditioned estimators overlap by two critical bands. The number of band-conditioned estimators is then $N-2$, which in our case is 13.

These experiments examine the contribution of appending neighboring bands to recognition accuracy. The information from all bands is off course available at merger input, but these experiments investigate if it is possible to obtain better estimation from band-conditioned probability estimator and consequently better final estimations and recognition performance.

The possible disadvantage of this technique is that it can degrade the noise robustness of the system. If a noise appears in one critical band, it affects more band-conditioned estimator outputs because of the frequency overlap introduced by this technique.

Merging of TRAP features from two bands – `2b_trap101_mvn`

TRAP features from two bands were concatenated creating band-conditioned probability estimator input.

Band-conditioned probability estimators structure: 202–300–29 (69600 weights)
Merger probability estimator structure: 406–300–29 (130500 weights)

Merging of TRAP features from three bands – `3b_trap101_mvn`

TRAP features from three bands were concatenated into one vector creating band-conditioned probability estimator input.

Band probability estimators net structure: 303–300–29 (99600 weights)
Merger probability estimator net structure: 377–300–29 (121800 weights)

The information about energy evolution in neighboring bands brings further improvement of the TRAP system performance as shown in Tab 4.3. The WER decreases almost one percent absolute when TRAP features from two bands are used as input to one band-conditioned classifier. The improvement obtained by adding TRAP features from another band is not significant over the two band input, but it still helps.

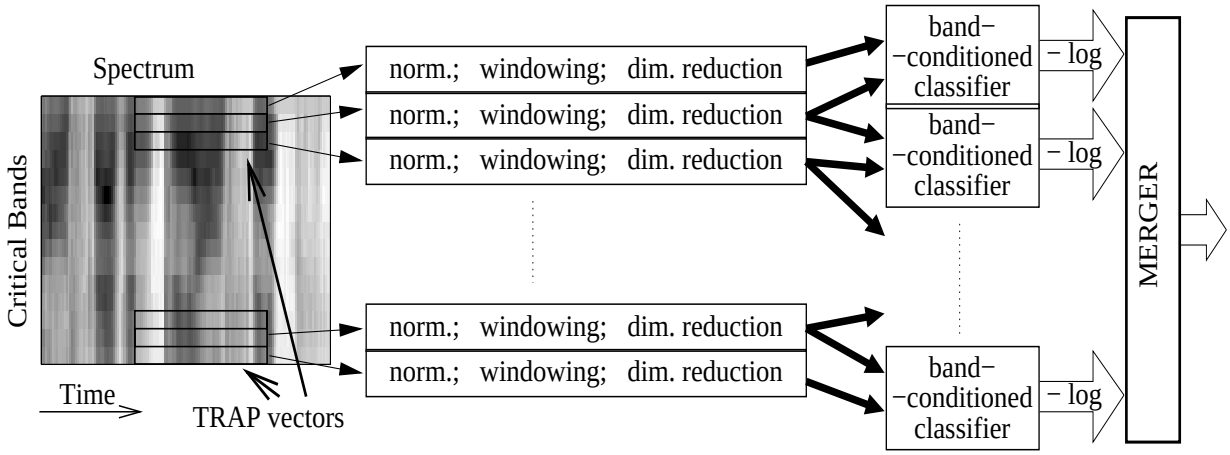


Figure 4.1: Band merging with independent processing of TRAP vectors

4.4 Band merging with dimensionality reduction

We would like to know if the dimensionality reduction has the same trends for vectors joined from more bands as it had for concatenated vectors from a single band.

There are two possibilities of dimensionality reduction – independent dimensionality reduction for each band and concatenation of the results (separate processing) or dimensionality reduction can be performed on concatenated vectors (joint processing).

4.4.1 Band merging after individual dimensionality reduction

Each TRAP vector is processed independently including the dimensionality reduction. TRAP features from neighbouring bands are then concatenated to create input for band-conditioned probability estimator. The block diagram of this technique is shown in the Fig. 4.1.

Merging TRAP-DCT features from two bands – `2b_trap101_sepDCT50_mvn`
50 DCT bases are applied to each TRAP feature vector to form TRAP-DCT features. The TRAP-DCT features from two consecutive bands are concatenated to create input into one band-conditioned classifier.

Band-conditioned probability estimators structure: 100–300–29 (38700 weights)
Merger probability estimator structure: 406–300–29 (130500 weights)

Merging TRAP-PCA feature from two bands – `2b_trap101_sepPCA50_mvn`
50 precomputed PCA bases are applied to each TRAP feature vector to form TRAP-PCA features. The TRAP-PCA features from two consecutive bands are concatenated to create input into one band-conditioned classifier.

Band-conditioned probability estimators structure: 100–300–29 (38700 weights)
Merger probability estimator structure: 406–300–29 (130500 weights)

Merging TRAP-LDA feature from two bands – `2b_trap101_sepLDA15_mvn`
15 precomputed LDA bases are applied to each TRAP to form TRAP-LDA features. Concatenated features from two consecutive bands create input into one band-conditioned classifier.

Band-conditioned probability estimators structure: 30–300–29 (17700 weights)
Merger probability estimator structure: 406–300–29 (130500 weights)

experiment	5^{th} band		merger		WER [%]
	train FA	CVFA	train FA	CVFA	
2b_trap101_sepDCT50_mvn	47.6	45.4	85.3	81.8	5.5
2b_trap101_sepPCA50_mvn	47.7	45.8	84.6	80.4	5.5
2b_trap101_sepLDA15_mvn	45.5	43.4	85.0	81.4	6.2
3b_trap101_sepDCT50_mvn	50.0	47.3	84.9	82.1	**5.1**
3b_trap101_sepPCA50_mvn	50.1	47.5	85.5	81.6	5.2
3b_trap101_sepLDA15_mvn	47.8	45.2	85.3	81.8	5.9

Table 4.4: Results for concatenating of separately processed TRAP vectors

Merging TRAP-DCT feature from three bands – `3b_trap101_sepDCT50_mvn`
50 DCT bases are applied to each TRAP feature vector to form TRAP-DCT features. The TRAP-DCT features from three consecutive bands are concatenated to create input into one band-conditioned classifier.

Band probability estimators net structure: 150–300–29 (53700 weights)
Merger probability estimator net structure: 377–300–29 (121800 weights)

Merging TRAP-PCA features from three bands – `3b_trap101_sepPCA50_mvn`
50 precomputed PCA bases are applied to each TRAP feature vector to form TRAP-PCA features. Features from three consecutive bands are concatenated to create input into one band-conditioned classifier.

Band-conditioned probability estimators structure: 150–300–29 (53700 weights)
Merger probability estimator structure: 377–300–29 (121800 weights)

Merging TRAP-LDA features from three bands – `3b_trap101_sepLDA15_mvn`
15 precomputed LDA bases are applied to each TRAP feature vector to form TRAP-LDA features. Concatenated features from three consecutive bands create input into one band-conditioned classifier.

Band-conditioned probability estimators structure: 45–300–29 (17700 weights)
Merger probability estimator structure: 377–300–29 (121800 weights)

The experimental results are shown in Tab 4.4. It can be seen that combination of dimensionality reduction and TRAP features concatenation techniques brings further improvement. This suggests that the techniques are complementary as it can be expected from the orthogonality of the processes in time-frequency plane.

By concatenating TRAP features from two bands on the input of band-conditioned classifier, significant improvement over one band input is achieved. Adding TRAP features from the third band brings further improvement.

4.4.2 Band merging followed by joint dimensionality reduction

In this scenario, the TRAP features are concatenated first and then the dimensionality reduction is applied. This approach may be beneficial mainly for data-driven techniques which can capture the dependencies over the bands. The block diagram of system with TRAP feature concatenation followed by joint dimensionality reduction is shown on Fig. 4.2.

The number of bases used in DCT and PCA joint dimensionality reduction was chosen to correspond to the number of bases used in separate processing – for concatenated TRAP feature vectors from two bands, the number of DCT and PCA bases was set to 100, and to 150 when feature vectors

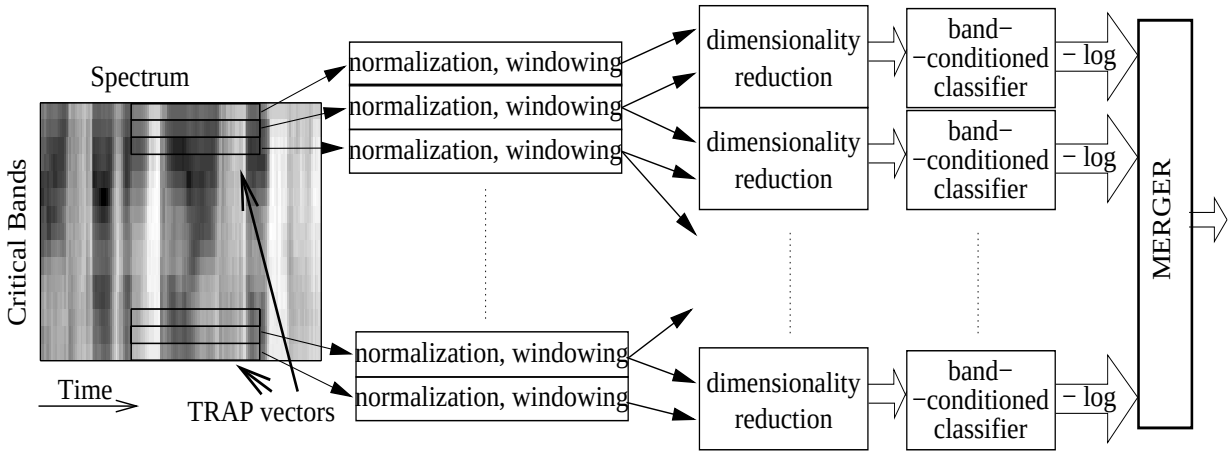


Figure 4.2: Band merging with joint dimensionality reduction of concatenated TRAP features

from three bands are concatenated. The number of LDA bases for two concatenated feature vectors was set to 15 because these bases are very noisy. The LDA bases for three concatenated vectors were not computed.

Merging TRAP features from two bands followed by DCT – `2b_trap101_joinDCT100_mvn`
100 DCT bases are applied on concatenated TRAP features from two consecutive bands. The resulting vector creates band-conditioned probability estimator input.

Band-conditioned probability estimators structure: 100–300–29 (38700 weights)
Merger probability estimators structure: 406–300–29 (130500 weights)

Merging TRAP features from two bands followed by PCA – `2b_trap101_joinPCA100_mvn`
100 PCA bases are computed and applied on concatenated TRAP features from two consecutive bands.

Band-conditioned probability estimators structure: 100–300–29 (38700 weights)
Merger probability estimators net structure: 406–300–29 (130500 weights)

Merging TRAP features from two bands followed by LDA – `2b_trap101_joinLDA15_mvn`
15 precomputed LDA bases are applied on concatenated TRAP features from two consecutive bands.

Band-conditioned probability estimators structure: 15–300–29 (13200 weights)
Merger probability estimators structure: 406–300–29 (130500 weights)

Merging TRAP features from three bands followed by DCT – `3b_trap101_joinDCT150_mvn`
150 DCT bases are applied on concatenated TRAP features from three consecutive bands.

Band-conditioned probability estimators net structure: 150–300–29 (53700 weights)
Merger probability estimators net structure: 377–300–29 (121800 weights)

Merging TRAP features from three bands followed by PCA – `3b_trap101_joinPCA150_mvn`
150 PCA bases are computed and applied on concatenated TRAP features from three consecutive bands.
For this experiment, it was also necessary to set the learning rate value (see Sec 3.4.1) to 0.4, otherwise the training would stop too soon with unsatisfactory accuracy.

Band probability estimators net structure: 150–300–29 (53700 weights)
Merger probability estimators net structure: 377–300–29 (121800 weights)

experiment	5^{th} band		merger		WER [%]
	train FA	CVFA	train FA	CVFA	
2b_trap101_joinDCT100_mvn	47.9	44.9	85.5	81.9	5.2
2b_trap101_joinPCA100_mvn	48.6	45.6	85.5	82.4	5.2
2b_trap101_joinLDA15_mvn	42.3	40.3	84.8	80.8	6.6
3b_trap101_joinDCT150_mvn	50.9	47.4	85.6	82.4	**4.8**
3b_trap101_joinPCA150_mvn	51.1	48.1	85.5	82.7	4.9

Table 4.5: Results for concatenating of TRAP features followed by dimensionality reduction

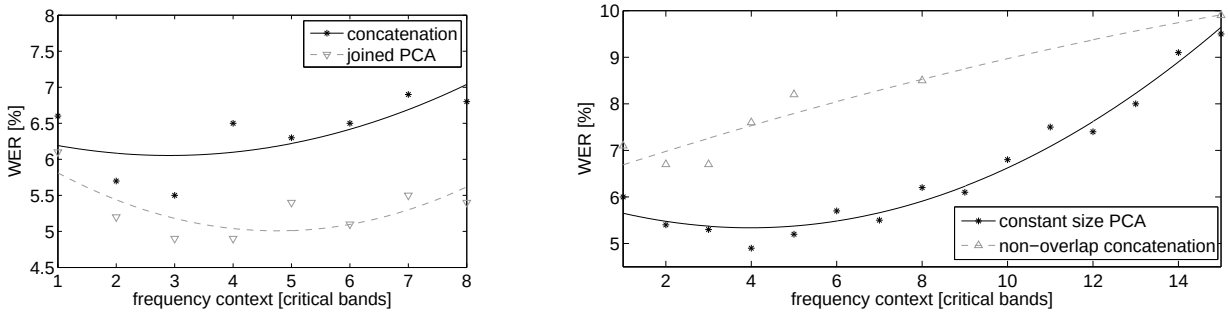


Figure 4.3: System performance when more bands are merged. Marked points are actual results, the lines are fits by second order polynomials

The results for concatenating TRAP features and processing them by joint dimensionality reduction are summarized in Tab. 4.5. We see that the LDA bases obtained form two concatenated TRAP vectors are not good for classification task and subsequent word recognition. As will be shown in the Sec. 4.6.2, these bases are very noisy.

On the other hand, the processing of concatenated TRAP features using DCT or PCA bases brings further improvement over concatenation of already processed TRAP vectors. This suggests that processing of concatenated vectors can capture inter-band information and use it for better classification. The dimensionality reduction bases will be further analyzed in Sec. 4.6.

4.5 Merging of more critical bands

In previous sections, it was shown that merging of TRAP features from few bands into one band-conditioned classifier input increases system performance. Question which can be asked is, whether adding more bands will lead to further improvement. Ultimately, we can put the whole block of critical band spectrogram as an input to a temporal classifier where only one neural net would be employed. For reducing the size of input vector to band-conditioned estimator, the above dimensionality reduction techniques can be used. So far, the size of the feature vector after joint DCT or PCA increases proportionally to the input. This can be however changed to fit different requirements. For example, the number of PCA bases can be chosen so that the percentage of preserved variability from input data is still the same, or that the size of output vector remains the same.

In Fig. 4.3, results of some experiments are shown. The ordinary concatenation of TRAP vectors with basic processing as it was introduced in Sec. 4.3 was tested with merging up to eight bands. The actual results are presented in the left picture by dark stars. The bright triangles represent performance of system where dimensionality reduction by PCA bases was applied on concatenated TRAP features as described in Sec. 4.4.2. The size of the output vector was half the size of input vector. The results of system where output dimensionality of concatenated TRAP features was kept

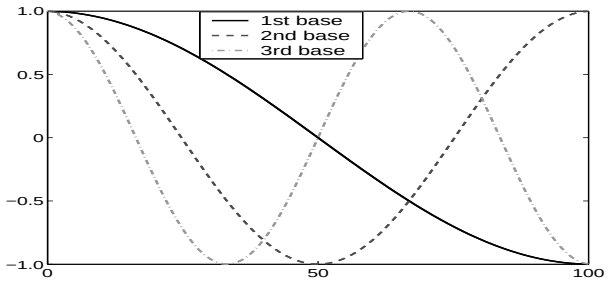


Figure 4.4: First three DCT base vectors

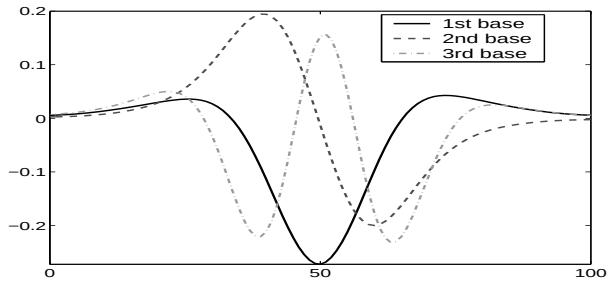


Figure 4.5: First three PCA base vectors

the same are depicted on the right picture of Fig. 4.3 by dark stars. The system is similar to above one with only difference in the number of used PCA bases for dimensionality reduction – 72 bases were used for all input sizes.

We also tested approach, where inputs to different band-conditioned estimators do not overlap, i.e. the frequency bands from which are estimated the conditioned class probabilities are separate. In this system the number of band-conditioned classifiers was $15/N$, N being the number of merged critical bands. In cases the fraction was not integer, the number of classifiers was the closest bigger integer and the remaining bands created input to the last classifier. Only ordinary concatenation of TRAP vectors with basic processing was used here, but the length of TRAP vector was 51 frames (actual frame ± 25 frames).

The curves in Fig. 4.3 are fits of given points by a second order polynomial. We see that most of the curves exhibit some mild minimum. The best performance of shown systems is obtained by merging three or four bands. Further merging do not bring more improvement and extremely wide inputs degrade the system performance. Merging of three bands thus looks like a good choice.

4.6 Dimensionality reduction bases

In this section we are taking closer look at the behavior of dimensionality reduction bases. Bases obtained from different critical bands by techniques are shown, analyzed and compared. Specially, we are interested in PCA bases obtained from concatenated TRAP vectors, because this processing outperformed the concatenation of single-band TRAP-PCA features on the input of band conditioned estimator.

4.6.1 Single band bases

1 band DCT

The DCT bases are given by an analytic equation (Eq. 4.2). The Fig. 4.4 is just illustrative and shows the first three DCT bases.

1 band PCA

The PCA bases were computed for each band (see section 4.2.2 for details). The first three PCA bases for 5^{th} band are shown in Fig. 4.5. It can bee seen that the PCA bases follow the property of DCT bases – the next base is "half period faster" than preceding one. The covariance matrix is shown in Fig. 4.6. We see that the the variance is highest in the center of the vector and is lower on the ends of the vector due to Hamming windowing. That is also why base vectors are going to zero towards the ends. The correlation between the frames is high for ± 3 frames and drops down for ± 10 frames, which corresponds to average phoneme length. Fig. 4.7 shows covered variability as function of the number of bases. We will cover 98% of the total variability by taking 41 bases. We cover 98.8% of the total variability by 50 vectors (used in experiments).

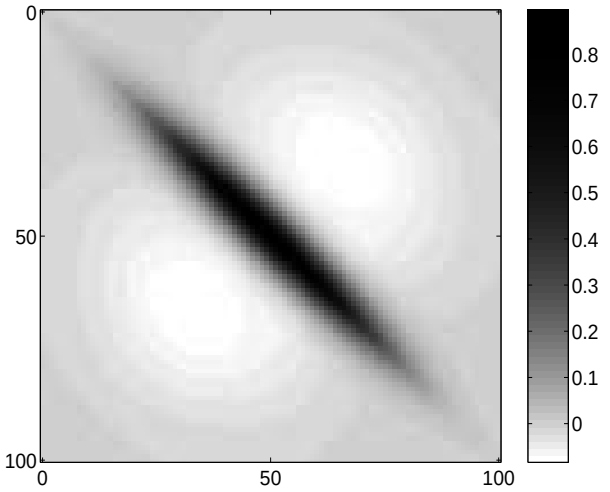


Figure 4.6: The covariance matrix for deriving PCA bases

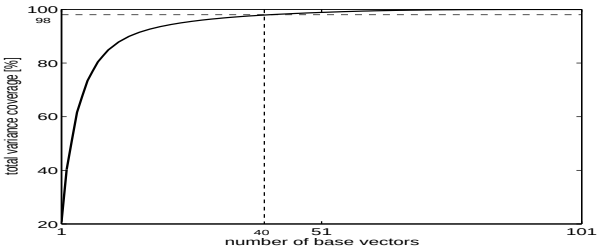


Figure 4.7: Variance coverage

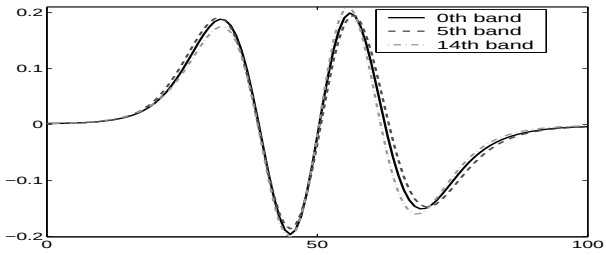


Figure 4.8: 4^{th} base vectors in different bands

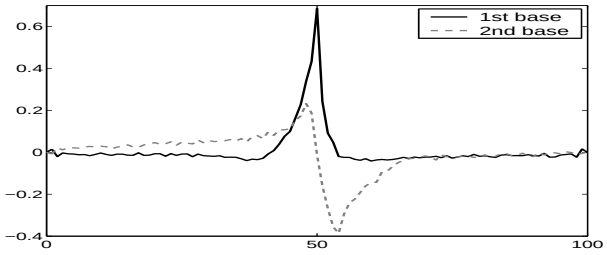


Figure 4.9: First and second LDA base vectors

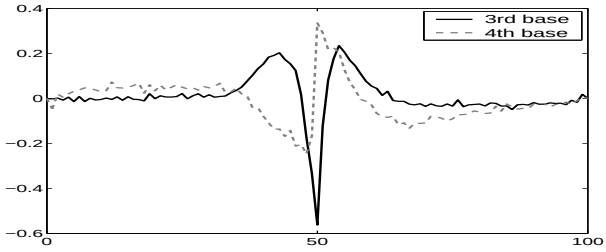


Figure 4.10: Third and fourth LDA base vectors

Fig. 4.8 shows the 4^{th} base vector for 0^{th}, 5^{th} and 14^{th} band showing almost no difference in base vectors over the bands.

1 band LDA

The LDA bases were precomputed as described in sections 4.2.3. The first and second base vectors are shown in Fig. 4.9, the third and fourth base vector are shown in Fig. 4.10. We see that the "active region" of LDA bases is much narrower than for PCA bases. The curves of LDA bases are also much sharper compared to relatively slow changes in PCA bases. We also see that the LDA bases are more "noisy". The difference of the base vectors over the bands is similar to that for PCA bases.

4.6.2 Concatenated band bases

2 and 3 band DCT

The DCT base vectors are given analytically so the shape is the same as in Fig. 4.4. They are wider for joined band processing.

2 band PCA

The PCA analysis is described in section 4.2.2. The bases are derived from the covariance matrix shown in Fig. 4.11. This example was estimated from concatenated vectors from 5^{th} and 6^{th} band. Fig. 4.12 shows the covered variability as function of the number of base vectors. We cover

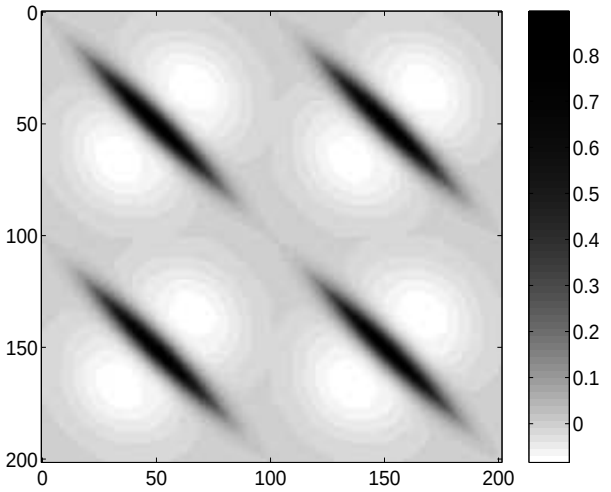


Figure 4.11: Covariance matrix of two joined bands

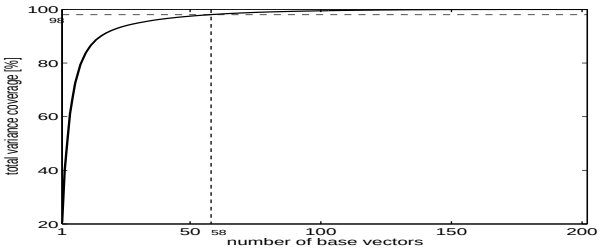


Figure 4.12: Variance coverage

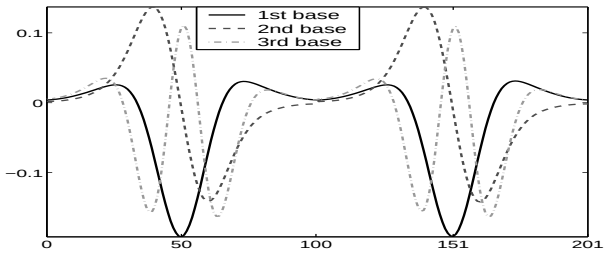


Figure 4.13: First three bases for two bands PCA

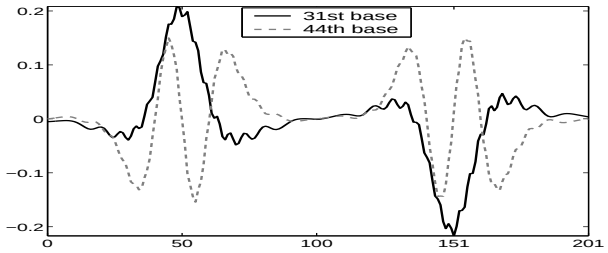


Figure 4.14: Base vectors with second half in opposite phase

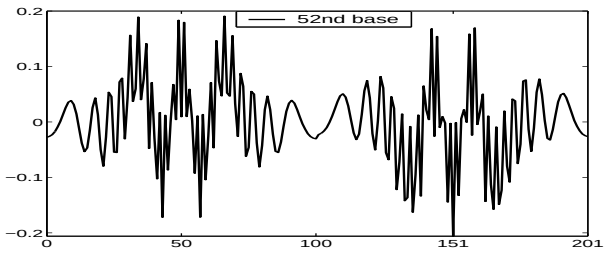


Figure 4.15: Basis vector with two frequency components

98% of the total variability by taking 58 base vectors. We cover 99.5% of the total variability by the 100 vectors (used in experiments).

The first three base vectors derived from this matrix are shown in Fig. 4.13. We can see that the bases vectors derived from concatenated TRAP features look like two concatenated base vectors derived from single band TRAP features. But some bases have different shape. The vectors with second half in opposite phase can be found. The examples of such a vectors are in Fig. 4.14. Some of these vectors have higher frequency component modulated on a low frequency shape. Examples of modulation is shown in Fig. 4.15. When base vectors are ordered according to their importance, the first one with second half in opposite phase is found on 31^{st} position. There are eight such vectors in first 58 bases (98% of total variability coverage) and 27 in first 100 bases. As for one band PCA, the base vectors are very similar for different bands.

2 band LDA

The LDA bases were precomputed as described in sections 4.2.3. The first and second base vectors are shown in Fig. 4.16, the third and fourth base vectors are shown in Fig. 4.17. We see that the shapes of LDA bases differ for the two concatenated vectors. Not only the shapes differ for different bands but also the dynamic range is very different. On contrary to one-band LDA base vectors, which were almost the same over the bands, the two band bases have very different shapes for different bands. The visible "noise" is present on the base vectors suggesting that there are problems with the estimation of **WC** and **AC** matrices. LDA vectors for more

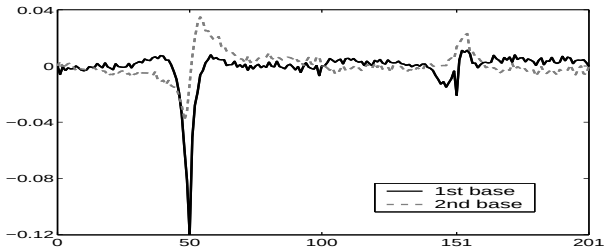


Figure 4.16: First and second base vectors for two bands LDA

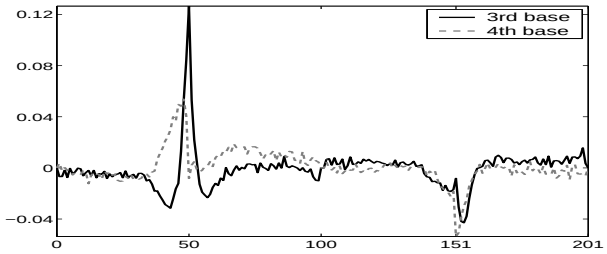


Figure 4.17: Third and fourth base vectors for two bands LDA

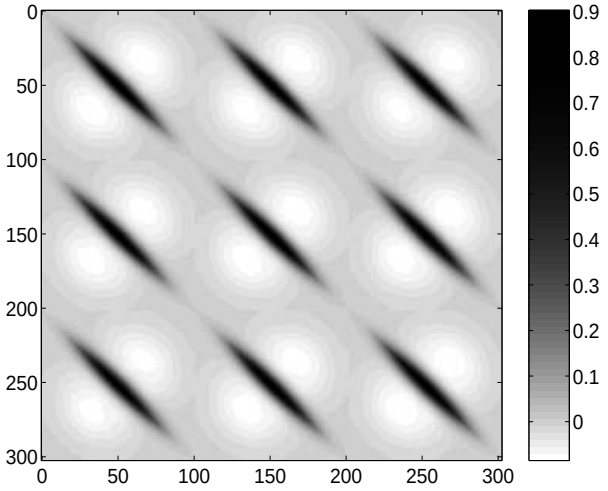


Figure 4.18: Covariance matrix of three joined bands

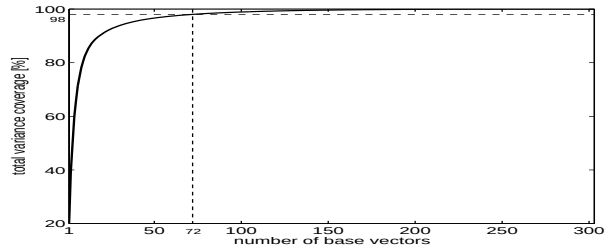


Figure 4.19: Variance coverage

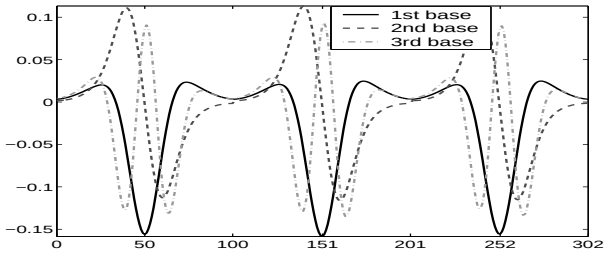


Figure 4.20: First three bases for three bands PCA

concatenated bands were not computed.

3 band PCA

The bases are derived from the covariance matrix shown in Fig. 4.18. This example was estimated from concatenated vectors from $4^{th}, 5^{th}$ and 6^{th} band. Figure 4.19 shows covered variability as a function of the number of base vectors. We cover 98% of the total variability by taking 72 base vectors. We cover 99.6% of the total variability by the 150 vectors (used in experiments). The first three base vectors derived from this matrix are shown in Fig. 4.20. We can see that the base vectors derived from concatenated TRAP features look like three concatenated base vectors derived from single band TRAP features. But also here, as for 2 band PCA, some bases have different shapes.

Besides the "integrating" base vectors shown in Fig. 4.20, there are the "differentiating" vectors shown in Fig. 4.21 and "accelerating" vectors shown in Fig. 4.22. Shown vectors could be artificially created by concatenation of corresponding 1 band bases. Thus 1^{st} 3 band PCA basis vector (Fig. 4.20) could be created as concatenation of three base vectors which are 1^{st} in single band PCA (Fig. 4.5) with coefficients $[1, 1, 1]$. In the same way, we could create the 19^{th} 3 band PCA basis vector (Fig. 4.21) with coefficients $[-1, 0, 1]$ and 77^{th} 3 band PCA basis vector (Fig. 4.22) with coefficients $[1, -2, 1]$. This simplification doesn't take into account actual values (amplitudes) of base vectors, but rather only their shapes. It can be also seen from

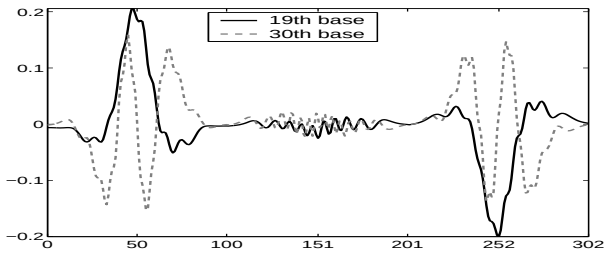


Figure 4.21: Differentiating shaped base vectors

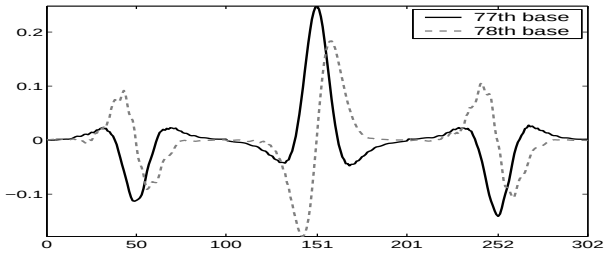


Figure 4.22: Acceleration shaped base vectors

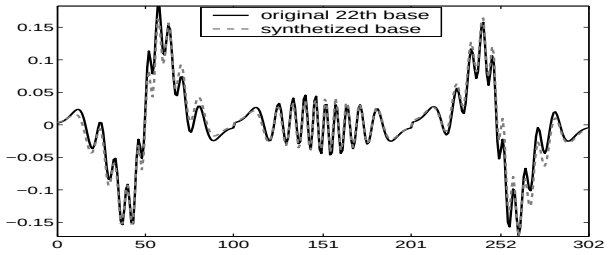


Figure 4.23: 22nd basis – original and synthesised

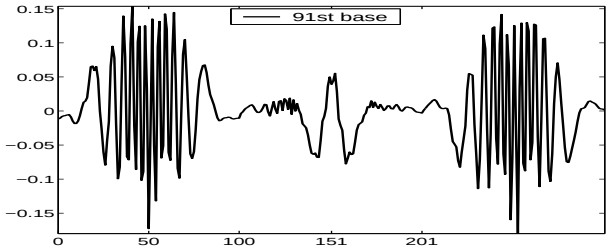


Figure 4.24: 91st basis with strong modulation by differentiating component

Fig. 4.21 and Fig. 4.22 that the actual shape of "differentiating" and "accelerating" vectors is more complicated as they have another component with higher frequency added (modulated) on their basic shape.

According to our observation and analysis, the bases with low frequency differentiating or accelerating shape are composed of sum of two or more bases. For example, we can synthesise the 22^{nd} base using two bases from 1 band PCA (described above). The 19^{th} base was taken in integrating manner with weight -0.22 and 2^{nd} base has differentiating manner and weight 0.66. The bases are shown in Fig. 4.23. The differences in the original and synthesised basis are quite small. The higher bases cannot be synthesised so easily, because different bases are used to create the resulting one. Fig. 4.24 shows the basis with strong accelerating and differentiating components and only weak integrating component.

4.6.3 PCA bases of more concatenated bands

The covariance matrices obtained from more than three concatenated TRAP feature vectors are proportional to that of two and three concatenated vectors. Again, the bases derived from concatenated TRAP features look like bases from single band TRAP features repeated several times. The differentiation – and also higher order derivatives – shapes are present in obtained bases. The more bands are merged, the sooner appear the basis capturing the cross-frequency variability. The first differentiation basis appears on 31^{st} place for two concatenated vectors, on 19^{th}, 17^{th} and 15^{th} for three, four and five concatenated vectors respectively. It points on increasing amount of variability in frequency dimension and the need to capture this variability. The example bases which capture the variability in frequency domain are shown in Fig. 4.25. Here, TRAP vectors with basic processing from five consecutive bands are concatenated. These bases capture changes in spectral energy across bands.

4.7 Summary

The presented results are summarized in Tab. 4.6. Apart from the systems using LDA for dimensionality reduction, other proposed modification of basic TRAP system brought improvement. We suspect,

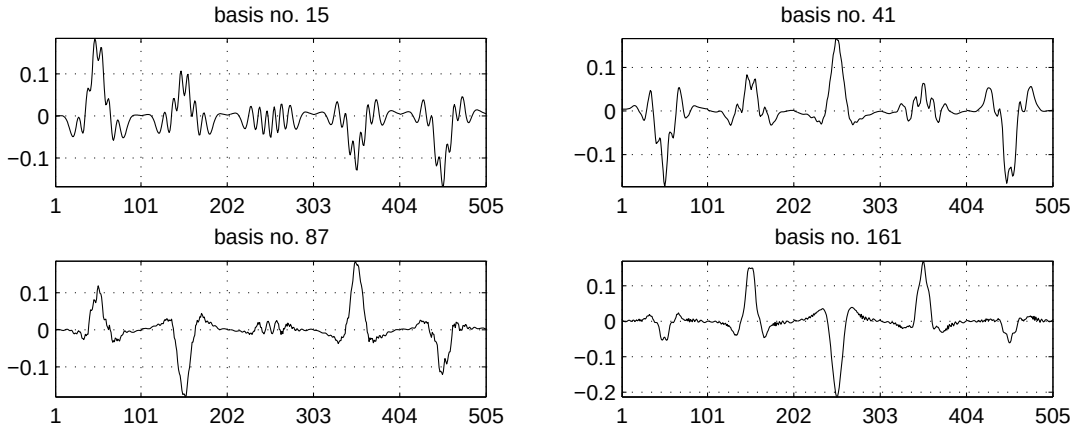


Figure 4.25: PCA bases capturing variability over frequency domain computed from concatenated TRAP vectors from five critical bands

No. of bands	1	2		3	
features		separate	joined	separate	joined
TRAP	6.6	5.7	—	5.5	—
TRAP-DCT	6.1	5.5	5.2	5.1	**4.8**
TRAP-PCA	6.1	5.5	5.2	5.2	**4.9**
TRAP-LDA	7.3	6.2	6.6	5.9	—

Table 4.6: WER of TRAP features derived from critical-band spectrogram

that the lower performance of the system which uses LDA is caused by the quality of LDA bases. For LDA computation, proper estimation of covariance matrices of vectors belonging to the same class is needed. This condition was probably not fulfilled with the amount of data used, resulting in noisy bases.

The systems with dimensionality reduction using DCT and PCA bases have the same performance. The DCT bases can be preferred over the PCA one because they do not need any additional computation. Although the goal of dimensionality reduction was to decrease the size of input to band-conditioned probability estimator, we also gained consistent improvement over the systems without dimensionality reduction.

When neighbouring bands are merged to create an input to band-conditioned classifier, significant improvement is reached. The dimensionality reduction using DCT or PCA bases of concatenated TRAP features brings further improvement over the system where dimensionality reduction precedes the concatenation. The best performing system gains almost 2% absolute better results than the TRAP baseline.

It was also shown that increasing frequency span of band-conditioned classifier beyond three critical bands does not bring further improvement and on the contrary, such increase can hurt the system performance.

Listening to poorly recognized files (as in section 3.2.1) was done also for these experiments and the recognition results were compared with transcriptions. Poor recognition was observed when:

- short file contains only one word not clearly pronounced. The sentence recognition accuracy oscillates between 100%, when recognized correctly, and 0%, when substituted by some other word.

- utterance contains background speech. The speech is most often clean but insertions are put where non-digit words occur.
- utterance contains speaker noise (like cough) at the ends of the utterance. This again causes insertion of additional digits in the recognized string.

Chapter 5

Modifications of critical band spectrogram

Two kinds of TRAP processing – dimensionality reduction and band merging – were studied in the previous chapter. It was shown that combination of both techniques brings large improvement. The study of data driven dimensionality reduction bases pointed out the importance of processing across bands. We have also seen that processing of more than tree bands does not bring further improvement. All these observations lead us to a question whether it is possible to modify critical band spectrogram in such a way, that it would replace the band merging and dimensionality reduction. Then the system would be the same as for TRAP baseline (Sec. 4.1) but the input to it would be a **modified critical band spectrogram**.

The modifying operators (MOs) are defined so that they locally modify the critical band spectrogram (CRBS) and create its modified version – modified critical band spectrogram (MCRBS). The TRAP-based probabilistic feature extraction is then done on MCRBS.

5.1 Modifying operators

One-dimensional modifying operators can modify the CRBS only in time or frequency domain. The bases used for dimensionality reduction of a single-band TRAP vector actually perform low-pass filtering of the critical band energy trajectory since the components capturing the higher frequencies are not used in the bases matrix. This suggests that the time filtering should have similar property. It can be done by MO which averages several frames in time domain.

The study of PCA bases of concatenated TRAP vectors has shown that the bases perform averaging and differentiation of the neighbouring bands. This can by done by an operator which averages or differentiates several frames across frequency domain.

The above mentioned one-dimensional MOs are accompanied with one performing differentiation across time domain. The context of the operators was set to one point on both sides of the central point, so in the frequency domain, three bands are processed as found optimal in the previous chapter. The operators can be seen in Tab. 5.1.

frequency average	frequency difference	time average	time difference
1 2 1	-1 0 1	1 2 1	-1 0 1

Table 5.1: One dimensional MOs – time and frequency averaging and differentiating

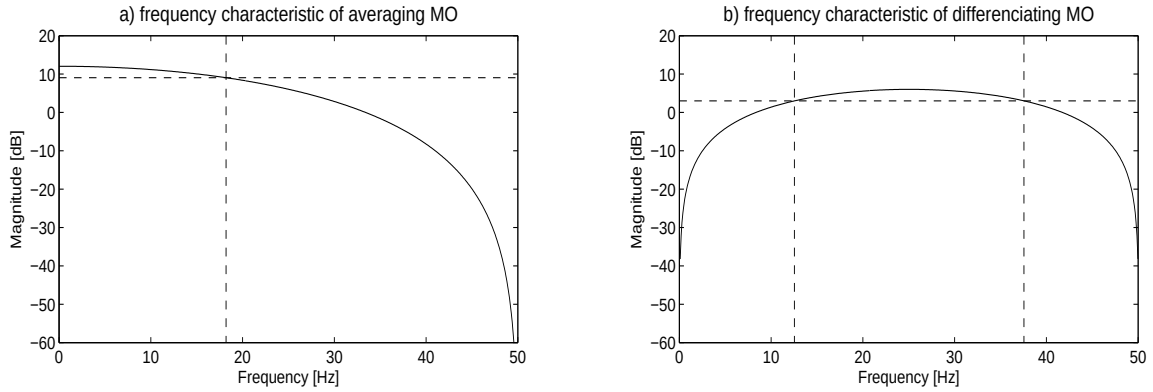


Figure 5.1: Frequency characteristics of modifying operators for 100Hz sampling

A MO works as a filter in given dimension. The frequency characteristic of averaging MO is shown in Fig. 5.1a. We see that the operator works as a low pass filter with cut off frequency around 18 Hz. The differentiating MO with frequency characteristic shown in Fig. 5.1b is a band-pass filter. The pass band is from 12 Hz to 37 Hz. The assumed sampling rate is 100 Hz as used for critical band energy estimation.

The one dimensional operators cannot cover all the aspects of band merging followed by dimensionality reduction. This can be better done by two-dimensional operators which are able to perform the modifications in both dimensions at the same time. Used two-dimensional MOs are 3×3 operators known as Sobel filters in image processing [44]. The operators are named **G1** to **G4** and their coefficients *g(t,f)* are shown in Table 5.2.

G1

-1	0	1
-2	0	2
-1	0	1

G2

1	2	1
0	0	0
-1	-2	-1

G3

0	1	2
-1	0	1
-2	-1	0

G4

-2	-1	0
-1	0	1
0	1	2

Table 5.2: Coefficients of two-dimensional G operators

The G1 operator performs averaging of bands, i.e. low-pass filtering over frequencies, and differentiation – band-pass filtering – in time domain. The G2 MO is just opposite to G1, it does differentiation across frequency and averaging across time. Operators G3 and G4 are harder to explain in terms of what they are doing in time and frequency domain. These operators capture changes in diagonal direction.

We compute the modified critical band spectrogram (MCRBS) as projection of the operator on the original CRBS. This operation is equivalent to the standard one- and two- dimensional FIR filtering with above shown filters. One point of modified CRBS in given (sampled) time t and in given frequency band f, $MCRBS(t, f)$ is:

$$MCRBS(t,f) = \sum_{i=-f_c}^{+f_c} \sum_{j=-t_c}^{+t_c} MO(i,j) \;\; CRBS(f+i, t+j) \tag{5.1}$$

where $MO(0,0)$ is the center point of the operator, f_c is frequency context of the operator and t_c is its time context. If we use one-dimensional operators, the context in the second direction will be zero. The effects of spectrogram modifications are shown in Fig. 5.2. The temporal patterns obtained from MCRBS are called "modified TRAP" – MTRAP.

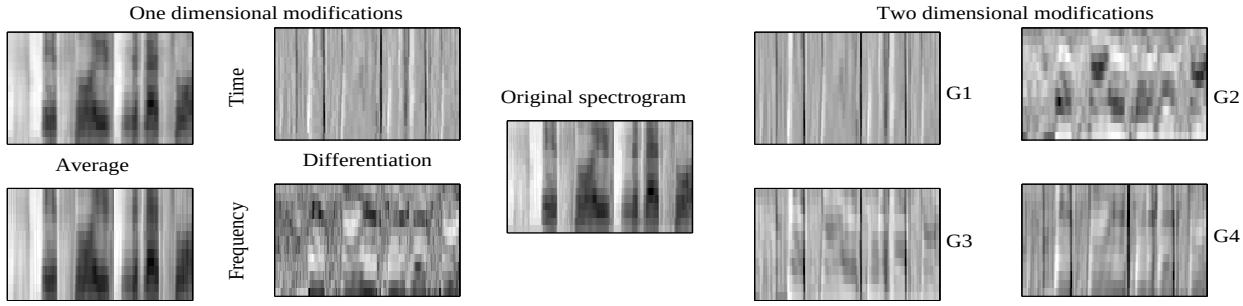


Figure 5.2: Modifications of critical band spectrogram

The size of resulting modified CRBS will differ from the size of original CRBS as the MCRBS is computed from several points of CRBS. To guarantee the same processing (i.e. length of TRAP vector) of both, modified and original CRBS, post-processing of MCRBS is needed to ensure obtaining probability estimates for each time slot of original utterance.

When CRBS with N frames is processed by MO with time context t_c, the resulting MCRBS will have $N - 2t_c$ frames. This problem can be overcome by two approaches:

- Artificially extend the original CRBS by t_c frames on each end so the resulting MCRBS will have the same number of frames as the original.
- Repeat the first and last frame of MCRBS t_c times to obtain the same number of frames as is in the original CRBS.

Since we have MOs with only 1 point of temporal context, the second approach is chosen and first and last frames of MCRBS are repeated.

Similarly, when CRBS with B bands is processed by MO with frequency context f_c, the resulting MCRBS will have $B-2f_c$ bands. Again the above mentioned approaches applied in frequency domain can guarantee the same number of bands as is in original CRBS. But this change is not critical for further processing. It means that the temporal classifier based on MCRBS with frequency processing will have less band-conditioned classifiers.

5.1.1 Experiments with modified critical band spectrogram

The experimental setup is described in chapter 3. The critical band spectrogram is modified by one of the introduced MOs and the system performance is evaluated. The results are shown in Tab. 5.3.

Time averaging/difference MTRAP – `TA_mtrap101_mvn; TD_mtrap101_mvn`

The original CRBS was modified by one-dimensional time averaging (TA) or time differentiating (TD) operator.

Band probability estimators net structure: 101–300–29 (39000 weights)
Merger probability estimators net structure: 435–300–29 (139200 weights)

Frequency average/difference MTRAP – `FA_mtrap101_mvn; FD_mtrap101_mvn`

The original CRBS was modified by one-dimensional frequency averaging (FA) or frequency differentiating (FD) operator.

Band probability estimators net structure: 101–300–29 (39000 weights)
Merger probability estimators net structure: 377–300–29 (121800 weights)

G operators MTRAP – `G#_mtrap101_mvn`

The original CRBS was modified by one of two-dimensional G operators. The number of operator will be placed instead of `#` in its work name.

experiment	5^{th} band		merger		WER [%]
	train FA	CVFA	train FA	CVFA	
baseline	43.0	40.0	84.5	81.5	6.6
TA_mtrap101_mvn	43.3	40.0	85.0	81.0	6.7
TD_mtrap101_mvn	43.5	40.2	84.3	81.3	6.6
FA_mtrap101_mvn	43.9	40.6	84.4	81.6	7.5
FD_mtrap101_mvn	38.0	35.8	86.3	78.9	7.1
G1_mtrap101_mvn	42.9	40.0	82.6	80.1	6.9
G2_mtrap101_mvn	35.3	33.1	85.6	78.0	7.5
G3_mtrap101_mvn	40.7	39.1	85.1	80.9	5.9
G4_mtrap101_mvn	43.4	41.0	86.0	81.2	6.3

Table 5.3: Results for MTRAP systems, baseline is repeated from Tab. 4.1 for comparison

Band probability estimators net structure: 101–300–29 (39000 weights)
Merger probability estimators net structure: 377–300–29 (121800 weights)

In Tab. 4.1, we see that none of the one-dimensional operators brought improvement over the TRAP baseline (Sec. 4.1). From two-dimensional MOs, only G3 operator, which is looking for diagonal changes in CRBS, improves over the TRAP baseline.

5.2 MTRAP with dimensionality reduction

Although the intention of spectrogram modification was to drop the dimensionality reduction from the TRAP processing, we may still consider the MTRAP vector as a signal which may be compressed and possibly enhanced by dimensionality reduction techniques. Specially the compression is possible because MCRBSs are obtained by filtering the original CRBS. As for the normal TRAP vector, the matrices of DCT and PCA bases are applied on MTRAP as it was introduced in sections 4.2.1 and 4.2.2.

Time averaging/difference MTRAP plus DCT

– `TA_mtrap101_DCT50_mvn; TD_mtrap101_DCT50_mvn`

The original CRBS was modified by one-dimensional time averaging (TA) or time differentiating (TD) operator. The DCT dimensionality reduction resulting in 50 point long vector was applied on the MTRAP vector.

Band probability estimators net structure: 50–300–29 (23700 weights)
Merger probability estimators net structure: 435–300–29 (139200 weights)

Frequency average/difference MTRAP plus DCT

– `FA_mtrap101_DCT50_mvn; FD_mtrap101_DCT50_mvn`

The original CRBS was modified by one-dimensional frequency averaging (FA) or frequency differentiating (FD) operator. The DCT dimensionality reduction resulting in 50 point long vector was applied on the MTRAP vector.

Band probability estimators net structure: 50–300–29 (23700 weights)
Merger probability estimators net structure: 377–300–29 (121800 weights)

G operators MTRAP plus DCT – `G#_mtrap101_DCT50_mvn`

The original CRBS was modified by one of two-dimensional G operators. The DCT dimensionality reduction resulting in 50 point long vector was applied on the MTRAP vector. The number of operator will be placed instead of `#` in its work name.

experiment	5^{th} band		merger		WER [%]
	train FA	CVFA	train FA	CVFA	
TA_mtrap101_DCT50_mvn	43.0	40.7	83.8	81.2	6.3
TD_mtrap101_DCT50_mvn	42.4	40.0	84.4	81.1	6.7
FA_mtrap101_DCT50_mvn	42.9	40.9	82.9	80.6	6.6
FD_mtrap101_DCT50_mvn	35.8	34.8	85.4	78.9	7.0
G1_mtrap101_DCT50_mvn	42.2	40.0	83.0	80.6	7.5
G2_mtrap101_DCT50_mvn	35.7	34.6	85.8	78.4	7.1
G3_mtrap101_DCT50_mvn	39.3	37.8	85.2	81.0	6.3
G4_mtrap101_DCT50_mvn	42.3	40.5	85.9	81.3	6.3

Table 5.4: Results for MTRAP systems with DCT dimensionality reduction

Band probability estimators net structure: 50–300–29 (23700 weights)
Merger probability estimators net structure: 377–300–29 (121800 weights)

Time averaging/difference MTRAP plus PCA

– `TA_mtrap101_PCA50_mvn; TD_mtrap101_PCA50_mvn`

The original CRBS was modified by one-dimensional time averaging (TA) or time differentiating (TD) operator. The dimensionality reduction using 50 PCA bases was applied on the MTRAP vector.

Band probability estimators net structure: 50–300–29 (23700 weights)
Merger probability estimators net structure: 435–300–29 (139200 weights)

Frequency average/difference MTRAP plus PCA

– `FA_mtrap101_PCA50_mvn; FD_mtrap101_PCA50_mvn`

The original CRBS was modified by one-dimensional frequency averaging (FA) or frequency differentiating (FD) operator. The dimensionality reduction using 50 PCA bases was applied on the MTRAP vector.

Band probability estimators net structure: 50–300–29 (23700 weights)
Merger probability estimators net structure: 377–300–29 (121800 weights)

G operators MTRAP plus PCA – `G#_mtrap101_PCA50_mvn`

The original CRBS was modified by one of two-dimensional G operators. The dimensionality reduction using 50 PCA bases was applied on the MTRAP vector. The number of operator will be placed instead of `#` in its work name.

Band probability estimators net structure: 50–300–29 (23700 weights)
Merger probability estimators net structure: 377–300–29 (121800 weights)

The results for system with MTRAP followed by DCT dimensionality reduction are shown in Tab. 5.4. Although the system performance changed, the difference is mostly not significant and the sign of performance change is different for each modification. The only significant difference to MTRAP system without dimensionality reduction is observed for frequency averaging (FA) MO.

The results for systems where PCA dimensionality reduction is applied on MTRAP vectors are shown in Tab. 5.5. A significant improvement over MTRAP system without dimensionality reduction was achieved for one-dimensional frequency operators and for two-dimensional operator G2. For the rest of the systems, the PCA dimensionality reduction has only a slight effect on system performance.

experiment	5^{th} band		merger		WER [%]
	train FA	CVFA	train FA	CVFA	
TA_mtrap101_PCA50_mvn	42.9	40.6	82.7	80.6	6.6
TD_mtrap101_PCA50_mvn	42.0	39.7	82.8	80.1	7.1
FA_mtrap101_PCA50_mvn	43.8	41.4	83.4	80.8	6.1
FD_mtrap101_PCA50_mvn	37.7	36.1	84.5	79.4	6.3
G1_mtrap101_PCA50_mvn	42.1	39.8	82.6	80.7	7.4
G2_mtrap101_PCA50_mvn	36.7	35.0	84.9	78.8	6.5
G3_mtrap101_PCA50_mvn	40.5	38.8	86.1	81.4	6.0
G4_mtrap101_PCA50_mvn	42.7	40.9	85.8	81.6	6.2

Table 5.5: Results for MTRAP systems with PCA dimensionality reduction

5.3 MTRAP Summary

The results for TRAP systems where features were derived from modified critical band spectrograms are summarized in Tab. 5.6. The table also contains the TRAP baseline results. Now now let us discuss each modifying operator:

Time averaging – TA – operator was inspired by the fact that the dimensionality reduction of the single TRAP vector actually implies low-pass filtering of the temporal trajectory. However, direct filtering of the trajectory by applying the operator does not have the same effect. It seems that the information needed for speech recognition is still preserved when the projection of TRAP vector on dimensionality reduction bases is done. When dimensionality reduction is applied on the MTRAP, the performance does not change significantly. This shows that MTRAP vector can be compressed without loss of information but also without the benefits seen for ordinary TRAP vectors.

Time differentiation – TD – operator performs band-pass filtering of the TRAP vector. MTRAP system with this operator performs the same as the baseline. This shows that the information about the speech events is not only in low-frequency components, but also in the higher ones. Dimensionality reduction applied on MTRAP degrades the system performance. This dimensionality reduction implies low-pass filtering which filters out part of the vital information.

Frequency averaging – FA – operator was inspired by the integrating property of the PCA bases obtained from concatenated TRAP vectors from three consecutive bands. But smoothing of the spectrogram in frequency direction, which this operator does, leads to degradation of the system. This operation can be also seen as obtaining the CRBS with wider critical bands (Sec. 2.2). This shows that frequency variability is very important and that such information should not be thrown away. Since the resolution in time direction was not changed, the dimensionality reduction of MTRAP brings improvement as it was seen on TRAP baseline system. MTRAP system with PCA dimensionality reduction has the same performance as the TRAP system with the same processing.

Frequency differentiating – FD – operator was also inspired by the shapes of PCA bases of concatenated TRAP vectors — some of them have the differentiating property. But again, performing only this operation on CRBS degrades the system performance. The dimensionality reduction of MTRAP by DCT bases does not have effect on system performance, when PCA bases are used, improvement is observed reaching similar performance as TRAP system with the same processing.

The observation of MTRAP systems with frequency averaging and differentiating and three band TRAP system with dimensionality reduction leads to a conclusion that individual type — i.e. integrating and differentiating — of bases cannot be applied on its own if system improvement is desired. Both properties are needed in order to improve the system performance. To confirm this hypothesis, a set of experiments was done. Conclusions from these experiments are given in the next section Sec. 5.3.1.

spectrogram modification	processing		
	basic	basic + DCT	basic + PCA
baseline	6.6	6.1	6.1
TA_mtrap101	6.7	6.3	6.6
TD_mtrap101	6.6	6.7	7.1
FA_mtrap101	7.5	6.6	6.1
FD_mtrap101	7.1	7.0	6.3
G1_mtrap101	6.9	7.5	7.4
G2_mtrap101	7.5	7.1	6.5
G3_mtrap101	5.9	6.3	6.0
G4_mtrap101	6.3	6.3	6.2

Table 5.6: WER [%] for systems based on modified CRBS

G1 operator combines frequency averaging and time differentiation. This operator seems to join the disadvantages of corresponding one-dimensional operators. The performance of this operator on its own is inferior to the baseline due to the smoothing of cross-band information. When dimensionality reduction is applied on MTRAP, the performance further degrades due to the loss of temporal information.

G2 operator combines frequency differentiation and time averaging. Also this operator shares the drawbacks of its one-dimensional components and the system performance with basic MTRAP processing is worse than the baseline. When dimensionality reduction is applied, the performance improves, but does not significantly outperform the baseline.

Operators G3 and G4 look for changes in diagonal direction. Both systems perform slightly better than the baseline. The dimensionality reduction does not have effect on the performance of systems with these operators.

The systems with some kinds of spectrogram modification achieve the performance of baseline system with dimensionality reduction. However, the performance of band merging technique was not reached. The results in Sec. 5.3.1 show that the complete information about the block of CRBS (i.e. spectral energy evolution in time and its frequency slope) is important to achieve good recognition performance. Complete information cannot be provided to the system by modifying the spectrogram by only one operator despite of the fact that they also process more critical bands. On the other hand, the performance of MTRAP systems is still reasonable, which means that significant portion of information needed for word recognition is present in each kind of modification.

To provide full information to the system, inputs from more modified CRBS are needed. This may be seen as drawback at first sight, but preprocessing the information in certain manner may help the system to concentrate only on certain aspect. Further combination of such partial information can be beneficial for overall system performance.

5.3.1 Performance of individual kinds of bases

A set of experiments was done to find out, what are the performances of different kinds of PCA bases used for TRAP dimensionality reduction. In experiments with MTRAP was observed, that processing the CRBS by individual modifying operators does not reach the same performance as processing the TRAP by PCA bases. Here we considered each bases kind independently to find out, whether the poorer performance of MTRAP systems is caused by only partial information presented to the system or improper design of the system. Also the different portions of bases of each kind in bases matrix were evaluated to see whether the optimal ratio exists.

Two comparative experiment were done first. Here the band-specific bases matrices were replaced

number of bases			WER [%]
int	diff	accel	
72	0	0	6.9
62	10	0	5.5
52	20	0	5.2
41	31	0	5.4
20	52	0	5.3
10	62	0	5.4
0	72	0	7.0

number of bases			WER [%]
int	diff	accel	
31	41	0	5.1
31	31	10	5.5
31	21	20	5.0
31	10	31	5.1
31	5	36	5.1
31	0	41	5.5
0	0	72	9.7

Table 5.7: WER [%] of system with dimensionality reduction of TRAP vectors from three consecutive bands by "manually" created bases matrix

by only one bases matrix used in all bands. This experiments answered the question whether individual handling of each band is necessary. The first comparative experiment was done for PCA dimensionality reduction of single band TRAP vector as it is described in Sec. 4.2.2. Second one was done for joined processing of concatenated TRAP vectors from three consecutive bands as it is described in Sec. 4.4.2. Both experiments give the same results with individual matrix for each band and common bases matrix used in all bands.

With this result, we could proceed further to "manual" creation of bases matrix. These bases were created as suggested in Sec. 4.6.2 part *3 band PCA* – single band basis was taken and concatenated three times with different amplitudes. Three kinds of bases were created:

integrating bases – they have the same shape and amplitude of individual components. One basis is created by concatenation of three one band vectors multiplied by coefficients [1, 1, 1].

differentiating bases – side components of these bases have the same shape but they are in opposite phase. The center component is zero. The vector is created by concatenation of three one band vectors multiplied by coefficients [-1, 0, 1].

accelerating bases – they have the same shape of individual components but they differ in phase and magnitude. The side components are in opposite phase to the center component and have half of its magnitude. One basis is created by concatenation of three one band vectors multiplied by coefficients [1, -2, 1].

Various sets of bases containing different portions of integrating (int), differentiating (diff) and accelerating (accel) bases were created. A subset of experimental results is given in Tab. 5.7. The WER of the system with 72 PCA bases is 5.3%. The results show that having a dimensionality reduction matrix made of one kind of bases only does not improve the system performance. But if other kinds of bases are added, the WER decreases and stays in a narrow interval of 5.0 – 5.5%.

Thus, we concluded that for improvement in system performance we need both: the actual evolution of critical band energy (possibly averaged over several bands) and the information about slope (or taperness) in the narrow frequency neighborhood. The MTRAP presented only certain kind of the information in the same way as the manually created bases and in such case, the improvement in system performance cannot be reached.

Chapter 6

Combinations of TRAP systems

In the previous chapter we have introduced modifications of critical band spectrogram. The spectrogram is modified by local one- or two- dimensional operator, which performs certain kind of filtering, to create its modified version. Each of the modified critical band spectrograms provides different information for the speech recognition system, which is actually only part of the information provided by the original spectrogram. The system thus focuses only on one aspect of critical band spectrogram behavior. It was shown that such system does not achieve performance competitive to other proposed techniques such as band merging with dimensionality reduction. To obtain more information from the original CRBS and thus achieve higher recognition accuracy, we can combine several MTRAP systems. Since MTRAP systems are trained on specific information, their combination should reach better WER than one system trained on unmodified CRBS.

The MCRBSs obtained by averaging and differentiating operators working in the same domain contain different information given by the nature of the filters which MOs represent. Far more interesting than combining differently filtered information in one domain is the combination of CRBS processed in different domains. Especially two-dimensional G operators create couples with perpendicular operators.

Not all possible combinations of MCRBS are tested. Rather, couples working in different domains are created. Also, combinations of four MCRBS obtained by all one- or two-dimensional MOs are tested.

Further we tested the combinations of original CRBS with MCRBS. These experiments should tell us whether focusing on one particular aspect of CRBS bring improvement over the system which sees only unmodified CRBS.

In our experiments, combined systems have the same (M)TRAP vector processing. Note that this condition is not necessary, but it restricts the experimental space to reasonable size. We do not seek the best results, for which more combination would need to be tested, but we want to verify possible advantages of combining specific information derived from the same source.

6.1 Multi-stream combination

Multi-stream combination technique combines the final probability estimations from different systems – i.e. outputs of merger probability estimators. If the individual systems are already trained, no additional training is needed for this kind of combination.

The outputs from the TRAP systems are posterior probabilities $P(q_k|\mathbf{x}_t, \theta)$, where the q_k is the k^{th} output class of total K classes, $\mathbf{x}_t$ is the input feature vector at time t and θ is set of probability estimator parameters. The systems have the same classes, thus we can use techniques for posterior probability combination. The resulting posterior probability vector for combining I systems will be $\hat{P}(q_k|\mathbf{X}_t, \mathbf{\Theta})$ were $\mathbf{X}_t$ is the set of all input vectors $\mathbf{X}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \dots \mathbf{x}_t^I\}$ and $\mathbf{\Theta} = \{\theta^1, \theta^2, \dots \theta^I\}$ is the set of all parameters.

combined spectrogram modifications	vector processing		
	basic	basic + DCT	basic + PCA
TA + FA	6.8	6.6	6.3
TA + FD	5.2	5.0	5.0
TD + FA	6.2	6.4	6.1
TD + FD	4.8	5.2	5.1
TA + TD + FA + FD	5.2	4.9	5.0
G1 + G2	5.1	7.5	5.2
G3 + G4	5.3	5.5	5.5
G1 + G2 + G3 + G4	4.9	4.6	4.8

Table 6.1: WER of multi-stream averaging combination of MTRAP systems

Base TRAP + modification	vector processing		
	basic	basic + DCT	basic + PCA
TA	6.7	6.1	6.4
TD	6.1	6.3	6.2
FA	7.0	6.4	6.1
FD	5.0	4.8	4.7
G1	6.5	6.4	5.9
G2	5.6	4.9	4.9
G3	5.3	5.1	5.2
G4	5.5	5.1	5.1

Table 6.2: WER of multi-stream averaging combination of base TRAP system and MTRAP system

6.1.1 Average of output probabilities

This technique simply averages the outputs belonging to the same class. The output probability for a given class k is the mean of particular probabilities:

$$\hat{P}(q_k|\mathbf{X}_t, \mathbf{\Theta}) = \frac{1}{I}\sum_{i=1}^{I} P(q_k|\mathbf{x}_t^i, \theta^i) \tag{6.1}$$

The results for multi-stream averaging combination of MTRAP systems are shown in Tab 6.1. Tab. 6.2 shows the results where base TRAP system was combined with one of the MTRAP systems.

6.1.2 Average of logarithm of output probabilities

This is another simple technique of multi-stream combination. The output probability for a given class k is the geometric mean of particular probabilities:

$$\hat{P}(q_k|\mathbf{X}_t, \mathbf{\Theta}) = \mathrm{e}^{\frac{1}{I}\sum_{i=1}^{I}\log(P(q_k|\mathbf{x}_t^i,\theta^i))} = \sqrt[I]{\prod_{i=1}^{I} P(q_k|\mathbf{x}_t^i, \theta^i)} \tag{6.2}$$

The results for multi-stream logarithm averaging combination of MTRAP systems are shown in Tab 6.3. The results where base TRAP system was combined with one of the MTRAP systems are shown in Tab. 6.4.

combined spectrogram modifications	vector processing		
	basic	basic + DCT	basic + PCA
TA + FA	6.8	6.0	6.4
TA + FD	4.6	**4.1**	4.7
TD + FA	6.1	6.4	5.9
TD + FD	4.5	4.7	4.9
TA + TD + FA + FD	4.8	4.3	4.7
G1 + G2	4.7	4.6	4.8
G3 + G4	4.8	4.6	5.0
G1 + G2 + G3 + G4	**4.2**	**4.3**	**4.3**

Table 6.3: WER of multi-stream logarithm averaging combination of MTRAP systems

Base TRAP + modification	vector processing		
	basic	basic + DCT	basic + PCA
TA	6.8	5.9	6.3
TD	6.1	5.9	6.0
FA	6.7	6.3	6.0
FD	4.5	**4.3**	**4.1**
G1	6.2	6.4	5.8
G2	4.7	**4.1**	**4.3**
G3	4.5	4.5	4.6
G4	4.8	5.0	4.8

Table 6.4: WER of multi-stream logarithm averaging combination of base TRAP system and MTRAP system

6.1.3 Entropy based combination

This new approach to multi-stream combination is described in [63]. The entropy of i^{th} system outputs at given time t:

$$h_t^i = -\sum_{k=1}^{K} P(q_k|\mathbf{x}_t^i, \theta^i) \log_2 P(q_k|\mathbf{x}_t^i, \theta^i) \tag{6.3}$$

can be used as its confidence measure. This information is used for weighting the outputs of combined systems.

High entropy means that the posterior probabilities are approaching equal probability for all classes. The system with high entropy has less discrimination, therefore outputs of such system should be weighted less. The system with low entropy has higher discrimination and its outputs should be weighted more. The weight for i^{th} system at time t is

$$w_t^i = \frac{1/h_t^i}{\sum_{i=1}^{I} 1/h_t^i} \tag{6.4}$$

The output posterior probability vector is then given by

$$\hat{P}(q_k|\mathbf{X}_t, \mathbf{\Theta}) = \sum_{i=1}^{I} w_t^i P(q_k|\mathbf{x}_t^i, \theta^i) \tag{6.5}$$

If the system entropy at given time is higher than a fixed (predefined) the threshold, then the entropy

combined spectrogram modifications	vector processing		
	basic	basic + DCT	basic + PCA
TA + FA	6.6	6.2	6.3
TA + FD	4.8	4.4	4.6
TD + FA	6.2	6.3	5.7
TD + FD	**4.3**	4.6	4.4
TA + TD + FA + FD	4.4	4.8	4.4
G1 + G2	4.8	4.7	4.8
G3 + G4	5.0	5.2	5.0
G1 + G2 + G3 + G4	**4.3**	**4.3**	**4.3**

Table 6.5: WER of multi-stream entropy based combination of MTRAP systems

Base TRAP + modification	vector processing		
	basic	basic + DCT	basic + PCA
TA	6.4	5.9	6.3
TD	5.9	6.0	5.9
FA	6.7	6.4	6.1
FD	4.6	**4.3**	**4.2**
G1	6.1	6.2	5.7
G2	5.1	4.5	4.4
G3	4.9	4.6	4.9
G4	5.0	5.1	4.8

Table 6.6: WER of multi-stream entropy based combination of base TRAP system and MTRAP system

is set to a large value to suppress the influence of this system. The modified equations are:

$$\tilde{h}_t = \begin{cases} 10000 & : \quad h_t^i > 1.0 \\ h_t^i & : \quad h_t^i \leq 1.0 \end{cases} \tag{6.6}$$

$$w_t^i = \frac{1/\tilde{h}_t^i}{\sum_{i=1}^{I} 1/\tilde{h}_t^i} \tag{6.7}$$

The **inverse entropy weighting with static threshold** was used in our experiments. The results for combination of MTRAP systems are shown in Tab 6.5. The results for combination of base TRAP system with MTRAP system are shown in Tab 6.6.

6.1.4 Summary of multi-stream combinations

First let us compare the results for individual combination technique:

- We can see improvement wherever the **frequency differentiation** is applied in either form – one-dimensional FD operator and two-dimensional G2 operators. So if we already have the information about temporal evolution of critical band energy, the information about frequency slope in its neighbourhood should be added as suggested in Sec. 5.3.1. Improvement can be also seen for combination of diagonal operators G3 and G4 over MTRAP systems with only one of them.

Base TRAP +	FD		G2	
processing	basic	basic + DCT	basic	basic + DCT
averaging	5.0	4.8	5.6	4.9
logarithm averaging	4.5	4.3	4.7	**4.1**
entropy based	4.6	4.3	5.1	4.5

Table 6.7: WER [%] of best performing systems with multi-stream combination

- The combinations of systems performing frequency averaging – FA and G1 operators – and base TRAP system or system with time modification – TA and TD operators – have similar performance as base TRAP system alone (see Tab. 4.6). This observation shows that frequency averaging does not provide new information useful for combination.
- The combination of all four systems which use one-dimensional operator does not bring further improvement. Its performance is close to the performance of frequency differentiation system combined with time modified system.
- The combination of all four systems which use two-dimensional operators brings further improvement over combination of two two-dimensional operators. This shows, that each G operator can extract vital information from original CRBS.
- The comparison between systems with various vector processing is not straightforward. When dimensionality reduction is applied, various effects discussed in Sec. 5.3 take place and their combination governs the resulting change in WER. The combination of systems with dimensionality reduction will have at least the same performance as the combination of systems with basic vector processing. This does not apply only for averaging combination of systems based on CRBS modified by G1 and G2 operator with DCT dimensionality reduction.

Comparing different multi-stream combination techniques, we see that best results are achieved by logarithmic averaging followed by inverse entropy weighting. The poorest performance is achieved with averaging combination. See Tab. 6.7 for overview of best performing systems.

The best results are obtained by logarithmic averaging of system with TRAP-DCT features and MTRAP-DCT features obtained from frequency differentiated CRBS.

6.2 Combination of band-conditioned classifiers outputs

The idea of band-conditioned classifiers combination is the following: if we have the estimates of class probabilities on the output of band-conditioned probability estimators, we can combine these probabilities. Each of band-conditioned classifiers can be seen as independent stream now and we can combine their outputs. However, these streams cannot be combined all together to create the final probability estimation directly. The accuracy of band-conditioned probability estimations is rather low and a nonlinear discriminative mapping is needed. Thus instead of merging probabilities estimated in critical bands into final probability estimation, certain kind of pre-processing of these probabilities is done. Merger will be trained on the pre-processed probabilities to estimate the final probabilities.

We added the pre-combination matrix to pre-process the outputs of band estimators and to form the input vector for merger probability estimator. The block diagram for two system combination with pre-combination matrix is shown in Fig 6.1a. The main task of pre-combination matrix is to combine only chosen conditioned estimates, which are now conditioned by band and system they came from, and to reduce the number of merger classifier inputs. All outputs belonging to the same class,

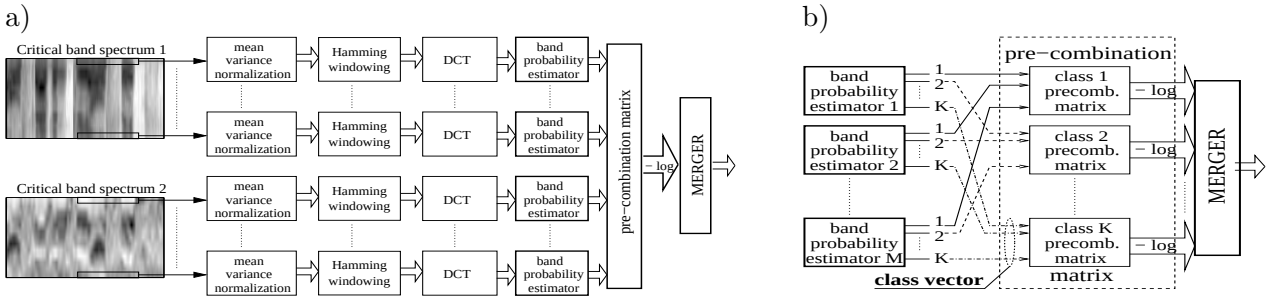


Figure 6.1: a) Block diagram of two system combination with pre-combination matrix
b) Detailed diagram of pre-combination matrix

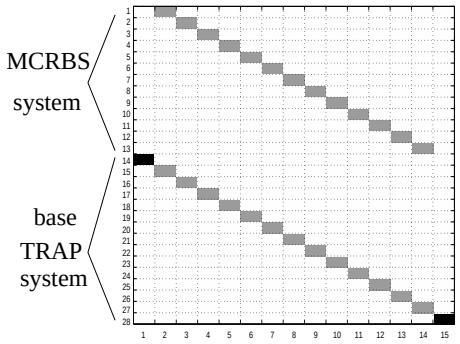


Figure 6.2: System averaging matrix

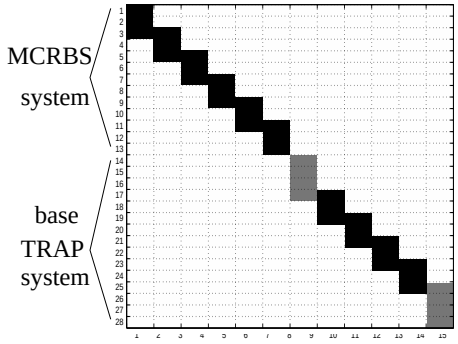


Figure 6.3: Band averaging matrix

which create **class vector**, are combined. Thus the pre-combination matrix can be seen as set of independent matrices, each one for one output class. The detailed diagram is shown in Fig. 6.1b. The pre-combined class vector (**pcv**) is computed as multiplication of class vector (**cv**) and its class pre-combination matrix (**CPM**):

$$\mathbf{pcv}_k = \mathbf{CPM}_k^T \times \mathbf{cv}_k$$

where $k = 1 \ldots K$ is the index of class. The input vector for merger probability estimator is created by concatenation of all pre-combined class vectors.

Note that in this section, only the results for the best combination from previous section – base TRAP system with frequency differentiating MTRAP – are presented. The TRAP vector processing used in the presented experiments is the basic processing, and basic processing with DCT dimensionality reduction (as in Tab. 6.7). This makes the comparison between proposed techniques easier without overwhelming the reader with tables and numbers.

6.2.1 Direct passing to merger

The band-conditioned estimators outputs without any pre-combination were passed directly to merger classifier first. This experiment tells us what is the capability of merger to extract information from different sources. Then we can compare, if the pre-processing is able to handle the partial information in a better way, i.e. achieve better word recognition accuracy. This method is very simple as it does not require any pre-processing. The disadvantage is increased size of the merger input vector.

6.2.2 Averaging pre-combination of the class vector

Then we conducted an experiment where outputs which belong to the same frequency band and different systems were averaged. We called this *system averaging*. Merging of partial probability

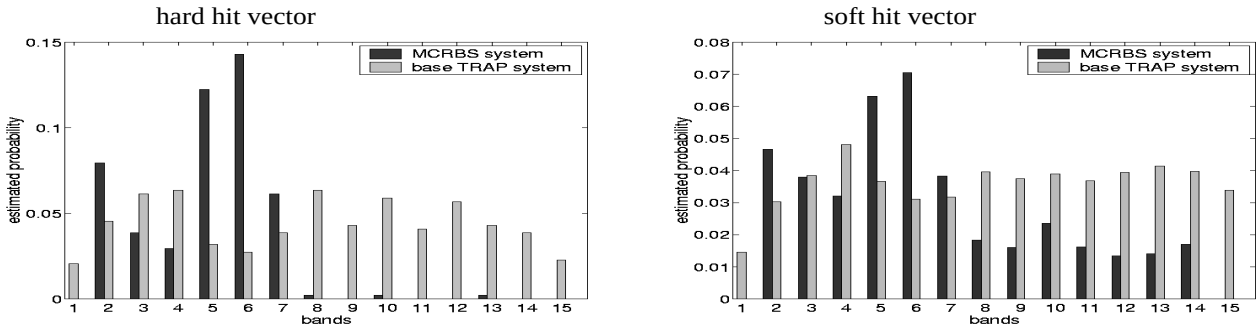


Figure 6.4: Hard and soft hit vectors

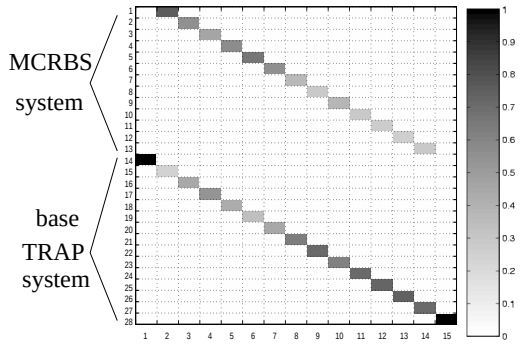


Figure 6.5: Weighted system averaging matrix

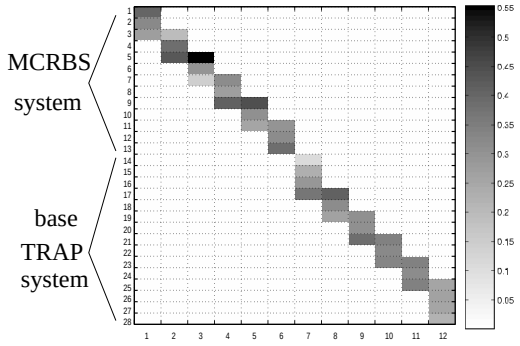


Figure 6.6: Weighted band averaging matrix

estimates from frequency bands to final estimates is left on the merger probability estimator as in the basic TRAP system.

In the subsequent experiment, in contrary to *system averaging*, the neighboring bands were averaged — *band averaging*. We suppose that the information from neighboring frequency bands from one system will be similar and thus can be merged. The merging of the information coming from different sources is left on the merger classifier. Three bands were averaged with one band overlap to obtain similar number of coefficients per class as in TRAP baseline. For base TRAP system, the probabilities obtained from first and last critical band, which usually have poor classification accuracy, are averaged together with three following/preceding bands. The examples of pre-combination matrices for these approaches are shown in Fig. 6.2 and Fig. 6.3

6.2.3 Weighted averaging pre-combination of the class vector

We also tried to emphasize more reliable outputs of band estimators in our experiments. For this, we performed the following analysis of the output of each band-conditioned estimator:

- Pass the data used for merger probability estimator training through the band-conditioned probability estimators and store the probability vector on the output of each estimator with corresponding label.
- Count how many times the class with the highest probability is equal to the label. Divide this number by number of occurrences of given label. We will call this *hard hit vector*.
- Add together the estimated probabilities of the same class as label. Divide this number by number of occurrences of given label. We will call this *soft hit vector*.

Now the performance for each output is known and can be used for emphasizing the outputs with higher accuracy. The weighted averaging pre-combination matrices are based on the hit vectors in the

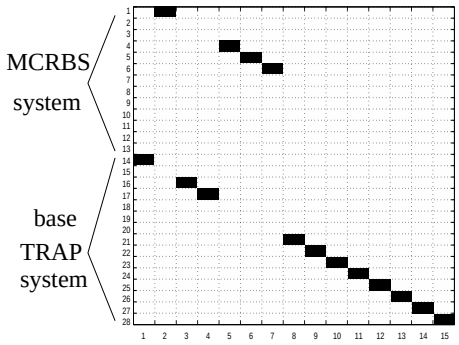


Figure 6.7: "Better system" pre-processing matrix

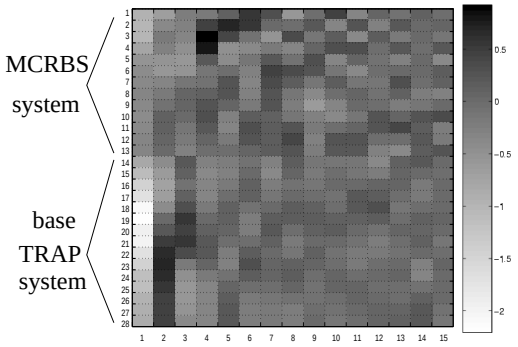


Figure 6.8: PCA pre-processing matrix

following way: The same structure of pre-combination matrices as in section 6.2.2, but each coefficient of the matrix is now weighted by the corresponding hit vector coefficient. The coefficients in matrices are "normalized" so that the sum of coefficients in each column is equal to one. The example of hit vectors for one label can be seen in Fig. 6.4. The pre-combination matrices derived from soft hit vectors are shown in Fig. 6.5, 6.6.

6.2.4 "Better system" pre-combination of the class vector

The extreme way of weighting is to take just the "better system" output. The coefficient for system which performs the best in given frequency band and for given class is set to one and the coefficient for the other system is set to zero. The decision about better system is based on hard hit vector. An example of "better system" pre-combination matrix is shown in Fig. 6.7. It corresponds to hit vectors shown in Fig. 6.4.

6.2.5 PCA pre-combination of the class vector

A data driven technique can be also used for pre-combination matrix design. The PCA was employed here. This technique has the ability to rotate the feature space in the direction of the largest variability. The coefficients with low variability do not carry information useful for further classification because their value is not changing over the data and thus cannot contribute to classification. Such coefficients can be omitted from output vector.

The PCA was computed for each class vector separately. Coherent behavior of the coefficients over the data is thus ensured and only directions important for given class can be found. The dimensionality of pre-combined class vector was fifteen points. The PCA bases vectors are in columns of the pre-combination matrix. An example of PCA pre-combination matrix for one class is shown in Fig. 6.8.

Note that the order of processing for this experiment differs from the common scheme shown in Fig 6.1a. Here, the negative logarithm is taken first and then the PCA computation or multiplication by the PCA bases is done.

6.2.6 Results

The results are summarized in Tab. 6.8. Only the results for the best of system combinations from previous sections are shown here – the combination of base TRAP system with frequency differentiating MTRAP. TRAP processing used in the presented experiments is the basic processing (mean and variance normalization followed by Hamming windowing) and basic processing with DCT dimensionality reduction.

Base TRAP +	FD		G2	
processing	basic	basic + DCT	basic	basic + DCT
no pre-combination	4.9	4.7	5.2	4.8
averaging - system	5.1	4.5	5.3	4.8
averaging - bands	5.0	4.6	5.0	4.6
W-h averaging - system	4.7	4.5	4.8	4.9
W-s averaging - system	5.3	4.7	5.0	4.6
W-h averaging - bands	4.6	5.0	5.0	4.6
W-s averaging - bands	4.9	4.7	5.1	4.5
better system	4.9	4.7	4.9	4.5
PCA	4.6	4.5	5.4	4.4

W-h — weighting according to hard hit vector;
W-s — weighting according to soft hit vector

Table 6.8: WER [%] of band-conditioned probability estimators outputs combinations

We can see that the results are very similar to each other and lie in 95% confidence interval. There is also no systematic behavior of combination techniques. One combination technique can give the best WER for one pair of systems and the worst for another. The WER of system where all outputs of band-conditioned classifiers were passed directly to the merger lies in the upper half of the observed interval. The reduction of merger input vector size by the combination of band-conditioned probability estimators outputs is possible without hurting the system performance but also without significant improvement. Actually only two pre-combination techniques – weighted band averaging according to the soft hit vector and better system – outperformed system with no pre-combination in all four cases.

6.3 Vector concatenation

The simple way to combine different streams is to directly concatenate the feature vectors and train only one classifier. The concatenation of the TRAP features is done on the input of band-conditioned probability estimator. It means that all processing (normalization, windowing, DCT) is done on each vector independently.

When vectors from critical band spectrograms with different number of bands have to be concatenated, the missing bands are replaced by repetition of the closest bands. For example we want to concatenate vectors from original CRBS and FD MCRBS, where FD MCRBS is missing first and last frequency band compare to the original CRBS. Then vectors for concatenation are taken from the second and last but one critical band respectively. Thus the final system has the same number of band-conditioned probability estimators as the system with more frequency bands (e.g. base TRAP system).

The band-conditioned probability estimators are trained on concatenated vectors. Since only specific part of information from original critical band spectrogram is presented to band-conditioned estimator, more accurate probability estimates should be obtained if the chosen part of information is relevant for classification. To find and learn dependencies between input vector and output class should be easier task for the classifier than to extract the relevant information from the input first and then learn the classification. If the band-conditioned classification is more accurate, the merger will provide better final estimates and subsequently better features for HMM system.

Fig. 6.9 shows the processing for system with vector concatenation. The results for this combination

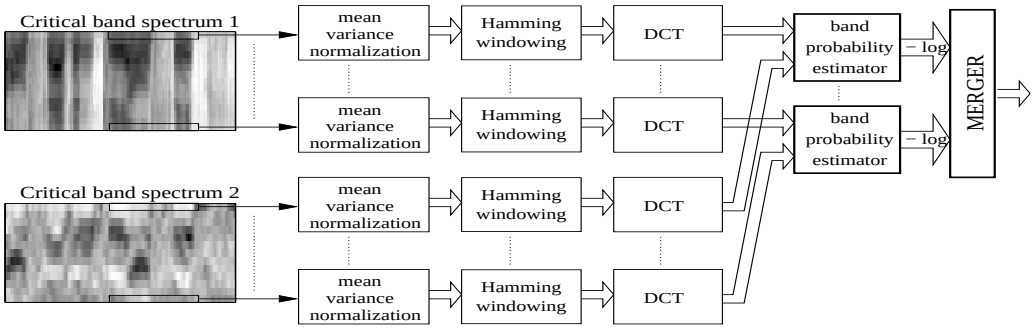


Figure 6.9: Block diagram of system with vector concatenation

Combined systems	processing	
	basic	basic + DCT
TA MTRAP + FA MTRAP	6.6	6.2
TA MTRAP + FD MTRAP	5.4	**4.4**
TD MTRAP + FA MTRAP	6.5	6.3
TD MTRAP + FD MTRAP	**4.9**	**4.4**
G1 MTRAP + G2 MTRAP	5.3	**4.4**
G3 MTRAP + G4 MTRAP	**5.0**	4.9
base TRAP + FA MTRAP	6.5	6.1
base TRAP + FD MTRAP	**5.1**	**4.4**
base TRAP + TA MTRAP	6.6	6.4
base TRAP + TD MTRAP	6.0	6.0
base TRAP + G1 MTRAP	6.3	6.2
base TRAP + G2 MTRAP	5.4	**4.2**
base TRAP + G3 MTRAP	5.4	5.0
base TRAP + G4 MTRAP	**5.0**	4.9

Table 6.9: WER [%] of vector concatenation combinations

approach are shown in Tab 6.9. The combination of two kinds of TRAP features were tested, all pairs from Sec. 6.1 were tested here.

Also for this kind of combination, the best results are obtained when frequency differentiating operator is used. There is also consistent improvement when TRAP vectors are transformed using DCT bases. The best performing system is the combination of base TRAP with G2 MTRAP with basic processing followed by DCT dimensionality reduction.

6.4 Summary

In this chapter, three techniques for combining different TRAP systems were introduced. These can be seen as different stages where the combination is done.

- The multi-stream combination technique was introduced in Sec. 6.1. It operates on the output of merger probability estimator – end of the TRAP-base probabilistic feature extraction.
- The combination of band-conditioned probability estimators outputs was introduced in Sec. 6.2 – it operates in the middle of probabilistic feature extraction.

Combination technique	Base TRAP + processing	FD basic	FD basic + DCT	G2 basic	G2 basic + DCT
Multi-stream	logarithm averaging	4.5	4.3	4.7	4.1
	entropy based	4.6	4.3	5.1	4.5
Band-conditioned estimators outputs	averaging - system	5.1	4.5	5.3	4.8
	averaging - bands	5.0	4.6	5.0	4.6
	better system	4.9	4.7	4.9	4.5
	PCA	4.6	4.5	5.4	4.4
Vector concatenation		5.1	4.4	5.4	4.2

Table 6.10: Best performing system combinations results overview – WER [%]

Combination technique	Multi-stream	band estimators outputs combin.	vector concatenation
Number of parameters without DCT	356 000	230 000	208 000
Number of parameters with DCT	326 000	200 000	178 000

Table 6.11: Number of parameters in combination system

- The combination technique which operates with TRAP features before nonlinear transformations was introduced in Sec. 6.3 and is called vector concatenation.

The overview of results for different combination is given in Tab. 6.10. This table is not complete, it rather gives comparison of the best performing methods used on the best system pairs – base TRAP + FD MTRAP and base TRAP + G2 MTRAP (as in tables 6.7 ans 6.8).

Better results are consistently seen for systems with DCT dimensionality reduction. These results thus once again confirmed the usefulness of this step.

The best results are obtained by multi-stream logarithm averaging combination. For TRAP vectors processed by DCT dimensionality reduction, the second best results are obtained by vector concatenation technique, but this technique does not work well with TRAP vectors with basic processing only. The combinations of band-conditioned estimators outputs do not achieve such good performance and are little behind the best system. However, the differences in performance of combined systems are rather small and do not exceed the 95% confidence interval from the average WER.

All combination techniques outperform the best results so far presented – system with band merging (Sec 4.3) reached 5.5% WER when three TRAP vectors after basic processing were presented at band-conditioned classifier input, and 4.8% WER when the concatenated TRAP features were processed by joint DCT dimensionality reduction (Sec 4.4.2). Thus it can be concluded, that when the partial information from the block of CRBS (three bands, 101 points) is derived first and proper combination technique is used, significant improvement over original system can be achieved.

The number of parameters in the whole system is another important issue. If we look at the proposed techniques from this point of view, we see that the multi-stream combinations need two whole systems, band-conditioned probability combinations need two times more band-conditioned classifiers and pre-combination matrix and vector concatenation need double size input to band-conditioned probability estimators. The approximate numbers of parameters for individual techniques are given in Tab 6.11. For our setup, the ratio between vector concatenation and multi-stream is 3:5. In case of using larger neural net for merger probability estimator, the ratio would go down to 1:2.

The results suggest that multi-stream system combination can be used when the partial systems are already trained or when there is a need to keep the information processing separate. However, when

suitable combination is found using multi-stream combination, the system with vector concatenation, which will have only half of the parameters and similar performance, can be trained.

Chapter 7

TRAPs and noisy speech

The performance of TRAP-based probabilistic features in noisy conditions will be examined in this chapter. We claimed that TRAP-based features should be less sensitive to noise, but so far all presented results were obtained on clean telephone speech. The databases used so far did contain some noise, as mentioned in section 3.2.1, but this noise is not controlled and it is not possible to evaluate feature performance related to noise. To be able to do so, we have to set up such experiment where noise conditions are varying and controlled.

7.1 Experimental setup

7.1.1 Used data

The AURORA2 database was designed to evaluate the speech recognition algorithms in noisy conditions. The framework was prepared as contribution to the ETSI STQ-AURORA DSR Working Group [70].

The source speech for this database is the TIDigits database [56], consisting of connected digits (11 word) spoken by American English speakers. A selection of 8 different real-world noises has been added to the speech with different signal to noise ratio (SNR). The noises are suburban train (suburb), crowd of people (babble), car, exhibition hall (exhibition), restaurant, street, airport and train station (train). Its levels are 20dB, 15dB, 10dB, 5dB, 0dB and -5dB.

The whole database is divided into two parts: training one and test one. The training part consists of 8440 utterances from 55 male and 55 female speakers. Two training conditions are defined:

- training on **clean** data only.
- training on clean and noisy – **multi-condition** – data. 20 splits are created. Each split contains few sentences from each speaker and one out of 20 noise scenarios is added to the signal. Noise scenarios are defined by 4 noises (suburb, babble, car, exhibition) at 5 SNRs (20dB, 15dB, 10dB, 5dB and clean).

Only the clean condition is used for training in our experiments. This simulates the situation when no noise (or test-specific) data are available during the training.

The test part consists of 4004 sentences from 52 male and 52 female speakers. There are 4 subsets with 1001 sentences. One noise with given SNR is added to each subset. There are three test sets:

- **Test A** – The added noises are suburb, babble, car and exhibition at all SNRs. The clean set is included. In total, this set consists of $4 \times (6 + 1) \times 1001 = 28028$ sentences. This test set contains the same noises as the multi-condition training (suburb, babble, car, exhibition) which leads to high match of multi-condition training and test data.

- **Test B** – The added noises are restaurant, street, airport and train at all SNRs. The clean set is included. In total, this set consists of $4 \times (6 + 1) \times 1001 = 28028$ sentences. This test set contains different noises than the multi-condition training showing influence of mismatch between multi-condition training and test data.

- **Test C** – The added noises are suburb and street at all SNRs. The clean set is included. In total, this set consists of $2 \times (6 + 1) \times 1001 = 14014$ sentences. The signals in this test set were filtered to simulate different frequency characteristics of the transmission channel which leads to high mismatch between multi-condition training and test data.

For more details about AURORA2 database see [71].

Next database used in our setup is OGI-Stories database [16]. It is used only in the neural net training part to enrich the variability of phoneme context.

7.1.2 HTK recognizer

The AURORA2 HTK reference recognizer [39] was used to evaluate proposed feature extraction techniques. The recognition task are strings of digits without restricting the string length. The digits are modelled as whole-word HMMs with the following parameters:

- 16 emitting states per word
- simple left to right models without skips
- mixture of 3 Gaussians per state
- diagonal covariance matrix

Two pause models are defined. The first, *sil*, should model pauses at the beginning and end of the sentence. It consists of three states modeled by mixture of 6 Gaussians each. The second, *sp*, should model pauses between words. It consist of a single state which is the same as the middle state of *sil* pause model.

The training is done in several steps by applying the Baum-Welch reestimation scheme.

The models are trained on clean training data only. This represents the case when no test specific (noisy) data are available for training. TestA and TestB then show the influence of different noises on the system. All conditions are actually mismatched as no noise is present during the training. TestC shows the behavior of the system when convolutional noise is also present in test data.

7.1.3 Neural net training

The training of neural net is done by Quicknet software as described in section 3.4.

The input critical band energies are computed in the following way:

- speech signal is divided into 25ms frames with 10ms shift (15ms overlap)
- power spectrum is computed from each frame
- power spectrum is filtered by 15 Bark scaled trapezoidal filters
- logarithm is taken

Targets for neural net training are represented by phoneme classes which occur in digits. The set contains 21 phonemes (in OGI-bet notation):
`ah ao ax ay eh ey f ih iy k n ow r s t th uw v w z sil` [1]

[1] For plosive phonemes, the closures were put together with release part.

label	index	AURORA2		Stories	
		time [s]	perc%	time [s]	perc%
ah	0	232.7	1.58	545.5	8.48
ao	1	488.4	3.32	135.4	2.10
ax	2	102.2	0.70	112.9	1.75
ay	3	1117.6	7.61	288.9	4.49
eh	4	261.3	1.78	224.5	3.49
ey	5	504.6	3.43	207.1	3.22
f	6	558.5	3.80	173.3	2.69
ih	7	366.3	2.49	383.8	5.97
iy	8	492.6	3.35	350.0	5.44
k	9	205.3	1.40	314.4	4.89
n	10	1007.2	6.86	425.8	6.62
ow	11	929.0	6.32	170.3	2.64
r	12	637.4	4.34	209.5	3.25
s	13	1008.1	6.86	514.6	8.00
t	14	555.2	3.78	469.8	7.30
th	15	274.2	1.87	53.0	0.82
uw	16	486.8	3.31	124.0	1.92
v	17	310.2	2.11	98.9	1.53
w	18	468.7	3.19	149.6	2.32
z	19	260.0	1.77	148.5	2.31
sil	20	4423.5	30.12	1327.3	20.65
total		14689.6		6427.1	

Table 7.1: Phoneme coverage in AURORA2 and OGI-Stories database

The AURORA2 database is not phonetically transcribed. To obtain target label for each frame, we performed the forced alignment of speech signals first. Models for forced alignment were trained on OGI-Stories database [16] using standard MFCC parameters. OGI-Stories database is also added into the neural net training set to enrich the phoneme context (in digits, phonemes are occurring in the same context), however, the cross validation part contains only sentences from AURORA2. The phoneme coverage in AURORA2 and OGI-Stories databases is shown in Tab. 7.1.

7.2 Selected experiments

In this section, the experimental results are presented. The performance of standard MFCC features is shown first, then we examine the performances of various TRAP-based probabilistic features. Not all experiments shown in chapters 4, 5 and 6 are repeated here. The experiments with good performance and improvement are selected and their behavior in noise is studied.

Each experiment has, beside its name, also a "work name" for further references. The structure of temporal probability estimators will be given in form: number of input units – hidden units – output units. We decided to keep the number of hidden units the same in all experiments. The band-conditioned estimators have 100 hidden units and the merging neural nets have 300 hidden units. The number of neural net weights for given probability estimator is given in brackets. The frame accuracies of the estimators are not shown, because they have only small informative value since the test data are different from training data.

The total number of word recognition results for one experiment under this setup is 70 (28 for

testA, 28 for testB and 14 for testC). The presentation of all numbers is not appropriate. Instead, the averages for given SNR and given test condition are given. Note, that the overall average word error rate is not average of the shown values – averages per test set – but an average of all experiments with the same SNR, which is a slightly different value.

7.2.1 MFCC baseline

The baseline features for AURORA2 database are 12 MFCC features (order 1 to 12) appended with logarithmic frame energy. The delta and acceleration coefficients are taken and together with direct features, they create 39 dimensional feature vector. The processing of speech signal to obtain MFCC coefficients is the following:

- Preemphasis with factor 0.97.
- Signal segmentation into 25ms frames with 10ms shift.
- Application of Hamming window and FFT.
- Binning of energies into 23 Mel filter bank.
- Logarithm and DCT computation.

noise	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
test A	0.7	6.3	18.7	40.6	64.4	83.1	93.0	43.8
test B	0.7	8.8	23.2	44.8	67.3	84.6	93.4	46.1
test C	0.8	9.1	18.1	34.7	60.8	83.8	90.5	42.5
average	0.8	7.9	20.4	41.1	64.8	83.9	92.7	44.5

Table 7.2: WER [%] for MFCC features

The results are shown in Tab. 7.2. MFCC features gain very good recognition accuracy for clean speech. The recognition performance degrade rapidly in presence of noise. The word error rate for the worst scenario (SNR = -5 dB) is similar to a random selection of words.

The test set B is the most difficult to recognize. We can see slight degradation in clean case for test set C which is due to the convolutional noise added into this test set. Test set C in noisy conditions seems to be easier to recognize than the test sets A and B. This behavior is given by the choice of noises for test set C. The WER for speech utterances corrupted with noises from test set C is lower than the average WER in test set A and B as shown in Tab 7.3.

test set	test A	test B	test C
subway noise	42.1	—	44.2
street noise	—	41.7	40.9
test set average	43.8	46.1	42.5

Table 7.3: Average WER [%] over all SNRs for noises from test C in all test sets

7.2.2 TRAP baseline – `AUR_base_trap101_mvn`

The baseline experiment has minimum processing of input critical band spectrogram and it is similar to Stories-Numbers baseline in section 4.1. The processing follows Fig. 2.6 and the steps to obtain baseline for TRAP-based probabilistic features are:

- 101 points of spectral energy in one critical band are taken
- mean and variance normalization of the 101 point vector
- Hamming windowing
- band-conditioned classifier training/forward pass
- concatenation of band-conditioned estimator outputs and negative logarithm
- merger classifier training/forward pass
- logarithm and decorrelation (KLT)

Band probability estimators net structure: 101–100–21 (12200 weights)
Merger probability estimators net structure: 315–300–29 (100800 weights)

noise	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
test A	1.8	6.7	11.2	19.6	38.9	69.9	87.9	33.7
test B	1.8	5.8	9.4	17.4	36.1	68.8	87.7	32.4
test C	1.9	7.4	12.3	21.8	38.9	67.5	86.9	33.8
average	1.9	6.5	10.7	19.2	37.8	69.0	87.7	33.2

Table 7.4: WER [%] for TRAP baseline features

The results for baseline TRAP based features are shown in Tab 7.4. We see that the best performance is achieved for clean test data and the WER increases with the SNR. If we compare the TRAP baseline with the MFCC baseline, we see that:

- TRAP baseline features have poorer performance for clean test data.
- The most difficult test set – B – for MFCC features turns into the easiest test set for TRAP-based features. This observation may suggest that the noises used in test set B are more frequency specific and TRAP-based features benefit from their architecture.
- The performance degradation of TRAP baseline system depending on SNR is slower compared to MFCC baseline. The performance of TRAP baseline system for SNRs in range from 15 to 5 dB is nearly two times better than the MFCC baseline.

Average recognition WER for noises from test set C and total test set averages are given in Tab. 7.5. We see that results for test set C noises are now much closer to the average WER compared to MFCC features. Also, the test set average WER are closer to each other. The order of test sets according to their recognition accuracies is B, A, C. The test set B is easier to recognize and the test set C is the most difficult. Also, the average WER for test sets A and C are very similar.

test set	test A	test B	test C
subway noise	33.2	—	34.2
street noise	—	33.1	33.4
total average	33.7	32.4	33.8

Table 7.5: average WER [%] for noises from test C in all test sets

	normalization	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
	no	1.6	10.2	26.9	58.3	83.5	90.8	92.4	51.9
sentence	mean	1.8	18.8	42.1	70.4	85.5	90.1	91.7	57.2
	mean and var.	1.9	8.1	13.8	24.5	44.9	72.1	88.9	36.3
vector	mean	1.8	15.9	38.4	70.0	87.2	90.7	91.5	56.5
	mean and var.	1.9	6.5	10.7	19.2	37.8	69.0	87.7	**33.2**

Table 7.6: WER [%] for base TRAP system with different normalization scene

7.2.3 Effects of normalizations

The modifications of short term cepstrum caused by Gaussian noise are reported in [69]. The authors identify three distinct modifications to the long-term statistics of the cepstrum caused by additive noise: mean shift, change of variance and distribution distorted from normal. The simple mean and variance normalization proposed by [59] which shifts and scales the distribution of cepstral coefficients helps – to a limited extend – in speech recognition of noisy speech. Since the cepstral coefficients are only critical band values transformed by linear discrete cosine transform, the same can by said for the critical band values.

So far, the mean and variance normalization of each TRAP vector was applied in all presented experiments. The effect of different kinds of normalization will be evaluated here. The noise levels presented in test set will show the sensitivity of given normalization to the noise.

The baseline `AUR_base_trap101_mvn` experiment is taken and the normalization is changed. The normalization can be done on whole sentence (file) or on each TRAP vector. Mean normalization, or mean and variance normalization can be done. Together with no normalization, there are five normalization scenarios.

The noise energy adds to the energy of speech (additive noise). Because we work with logarithmic energy, this addition is more visible in low energy regions – silence. The regions with high energy (for example voiced phonemes) are distorted less. As the background log-energy increases and the log-energy of speech changes only slightly, the temporal trajectory of critical band energy will start at higher value and will have smaller dynamics.

If no normalization is done, the parts of noisy speech signal which are not affected by noise – high energy regions – can be correctly classified and recognized. But TRAP techniques work with long context so the values around the current frame also contribute to its classification. As result, TRAP based features with no normalization are inferior to MFCC features which take in account only the current frame and the dynamics from short context.

Mean normalization without the variance one degrades the performance of the system. Due to the mean normalization, the values from low- and high- energy regions of noisy speech signal are shifted and the classification and recognition suffer.

The mean and variance normalization preserves the dynamics of the vector, however, the values are not identically preserved. The overall shape of the vector is more important for correct classification than the absolute values.

The results are shown in Tab 7.6. They indicate, that if training and test conditions are matching (clean test data), the best results are obtained without any normalization. This changes for noisy data which were not observed during the training. Here, the best performance is achieved when mean and variance normalization is done over one vector. Therefore the following experiments will be based on mean and variance normalized TRAP vectors.

7.2.4 TRAP dimensionality reduction (DCT, PCA)

The dimensionality reduction – matrix multiplication – is placed between the normalization and band-conditioned estimator input as described in section 4.2. Here, the matrices consisting of DCT or PCA bases are tested.

TRAP plus DCT – `AUR_base_trap101_DCT50_mvn`
50 cosine bases were used for dimensionality reduction.

TRAP plus PCA – `AUR_base_trap101_PCA50_mvn`
50 PCA bases were used. These bases were computed for each band on the band probability estimator training data (clean training AURORA2 + Stories) and stored. For the other data parts, these precomputed base vectors were used.

Band probability estimators net structure: 50–100–21 (7100 weights)
Merger probability estimators net structure: 315–300–29 (100800 weights)

noise	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
test A	1.7	5.7	9.7	18.8	41.6	75.9	89.7	34.7
test B	1.7	5.1	8.6	17.1	42.1	76.5	90.0	34.4
test C	1.5	6.2	10.7	20.1	42.5	74.5	89.5	35.0
average	1.6	5.6	9.5	18.4	42.0	75.9	89.8	34.7

Table 7.7: WER [%] for TRAP with DCT dimensionality reduction

noise	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
test A	1.6	6.0	11.1	21.9	45.5	75.8	89.2	35.9
test B	1.6	5.3	9.6	19.7	44.0	74.5	88.6	34.8
test C	1.5	7.2	11.9	23.5	46.0	74.1	87.8	36.0
average	1.6	6.0	10.7	21.3	45.0	74.9	88.7	35.5

Table 7.8: WER [%] for TRAP with PCA dimensionality reduction

The results for TRAP systems with dimensionality reduction are shown in Tab. 7.7 and Tab. 7.8. The dimensionality reduction technique brings improvement in case of clean test data and in weak noise with SNR 20 dB. For noises with SNR 15 dB and less, these systems degrade faster than the TRAP baseline. The system where PCA bases were used for dimensionality reduction is more sensitive to noise than the system were DCT bases were used. It can be explained by the fact that PCA is computed from the data. Only clean data are used for PCA computation. The bases are then used also on noisy data, so it is one more data-sensitive processing step in the TRAP-based feature extraction.

7.2.5 Band merging without and with dimensionality reduction

Band merging is a technique, which concatenates TRAP vectors from neighboring critical bands on the input of one band-conditioned probability estimator. The information from these bands is merged into one estimator output. Improvement was seen when applying this technique on clean data. See section 4.3 for further details.

Here, we test systems where 3 bands create an input to one band-conditioned probability estimator. These bands are taken with one band shift, resulting in 13 band conditioned probability estimators. First, system without dimensionality reduction is tested. When dimensionality reduction is incorporated into the system, both independent and joint processing of TRAP vectors are tested.

noise	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
AUR_3b_trap101_mvn	1.8	5.5	9.8	20.3	42.5	74.3	89.4	34.8
AUR_3b_trap101_sepDCT50_mvn	1.4	5.0	10.1	22.8	52.2	82.2	90.2	37.7
AUR_3b_trap101_sepPCA50_mvn	1.5	5.0	9.4	22.2	53.1	83.1	90.3	37.8
AUR_3b_trap101_joinDCT150_mvn	1.5	5.4	11.2	26.6	56.7	83.0	90.4	39.3
AUR_3b_trap101_joinPCA150_mvn	1.4	5.7	12.7	32.3	69.0	88.5	91.7	43.1

Table 7.9: Average WER for band merging without and with dimensionality reduction

3 band TRAP – AUR_3b_trap101_mvn
TRAP vectors from three bands are concatenated into one vector creating band-conditioned probability estimator input. No dimensionality reduction is applied.

Band probability estimators net structure: 303–100–21 (32400 weights)
Merger probability estimators net structure: 273–300–21 (88200 weights)

3 band TRAP plus separate DCT – AUR_3b_trap101_sepDCT50_mvn
50 DCT bases are applied separately on each TRAP vector from three consecutive bands. The resulting vectors are concatenated.

3 band TRAP plus separate PCA – AUR_3b_trap101_sepPCA50_mvn
50 PCA bases are applied separately on each TRAP vector from three consecutive bands. The resulting vectors are concatenated. The PCA bases are computed for each band on the band specific data (clean training AURORA2 + Stories)

3 band TRAP plus joint DCT – AUR_3b_trap101_joinDCT150_mvn
150 DCT bases are applied on concatenation of TRAP vectors from three consecutive bands.

3 band TRAP plus joint PCA – 3b_trap101_joinPCA150_mvn
150 PCA bases are applied on concatenation of TRAP vectors from three consecutive bands. The PCA bases are computed for each band-conditioned estimator on its specific data (clean training AURORA2 + Stories)

Nets structure for systems with dimensionality reduction

Band probability estimators net structure: 150–100–21 (17100 weights)
Merger probability estimators net structure: 273–300–21 (88200 weights)

The results for all systems with band merging are shown in Tab 7.9.

When looking at results for system with band merging and without dimensionality reduction – AUR_3b_trap101_mvn – the improvement over the baseline can be seen for clean case and weak noise with SNR 20 dB. But the performance degrades below the baseline for noises with SNR 15 dB and stronger. This observation is in agreement with the claim in section 4.3 – the noise is affecting more band estimators when TRAP vectors from several bands are concatenated and thus impairs the recognition accuracy.

The results for systems with band merging and dimensionality reduction performed on each critical band independently are presented next. Here again, the systems improve in clean case or weak noise cases and degrade when strong noise occurs in the test signal. The system which uses PCA bases for dimensionality reduction is slightly inferior to the one using DCT bases.

The results for systems with band merging and dimensionality reduction on concatenated TRAP vectors from three bands are shown in rows titled AUR_3b_trap101_joinDCT150_mvn and AUR_3b_trap101_joinPCA150_mvn. The improvement over 3 band system without dimensionality reduction is achieved only for clean test data. The systems have similar performance for weak noises

noise	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
AUR_FD_mtrap101_mvn	2.1	7.6	15.2	33.1	63.1	84.5	90.8	42.3
AUR_G2_mtrap101_mvn	2.3	7.9	14.2	28.7	54.0	78.9	89.6	39.4
AUR_G3_mtrap101_mvn	1.6	7.4	12.3	23.4	45.4	74.0	88.3	36.1
AUR_G4_mtrap101_mvn	1.8	6.5	12.0	24.2	47.7	75.2	88.1	36.5

Table 7.10: average WER of MTRAP systems

noise	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
AUR_FD_mt101_DCT50_mvn	2.0	8.4	15.7	32.1	59.4	81.9	89.6	41.3
AUR_G2_mt101_DCT50_mvn	2.0	8.3	15.8	32.2	59.7	82.5	90.1	41.5
AUR_G3_mt101_DCT50_mvn	1.8	7.6	12.6	23.2	44.1	73.2	89.6	36.0
AUR_G4_mt101_DCT50_mvn	1.6	6.9	12.1	23.7	45.7	73.9	88.6	36.1

Table 7.11: average WER [%] of MTRAP systems with DCT dimensionality reduction

noise	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
AUR_FD_mt101_PCA50_mvn	1.9	7.7	16.7	37.5	65.9	84.5	90.3	43.5
AUR_G2_mt101_PCA50_mvn	2.0	7.8	16.1	35.3	65.4	85.3	90.5	43.2
AUR_G3_mt101_PCA50_mvn	1.8	7.1	11.8	22.7	45.0	73.0	88.1	35.6
AUR_G4_mt101_PCA50_mvn	1.7	6.1	11.8	24.8	49.7	77.1	88.7	37.1

Table 7.12: average WER [%] of MTRAP systems with PCA dimensionality reduction

and degrade for strong noises. Compared to TRAP baseline, these systems bring improvement only in clean case and noise with SNR 20dB. The degradation grows with SNR worse than 20dB. The system which uses joint PCA bases for dimensionality reduction has the poorest performance. It degrades faster with increasing noise level than any other system. This observation supports the hypothesis, that the PCA is another noise sensitive part of the system which has a big impact on system performance (compare with system which uses DCT bases for dimensionality reduction).

7.2.6 MTRAP systems

Chapter 5 describes possibilities of critical band spectrogram modifications and TRAP systems based on these modified critical bans spectrograms (MCRBS). Some of these systems were tested on noisy speech data. The tested systems can be divided into two groups:

- systems which bring improvement in combination with other systems or TRAP baseline – these are using modifying operators (MO) G2 or FD.
- systems which had good performance themselves – these are using modifying operators G3 or G4.

All mentioned MOs are using 3 critical bands from original spectrogram resulting in 13 bands in MCRBS. For details about spectrogram modification see Section 5.1.

The experiment names are `AUR_MO_mt101_mvn`, where `MO` is one of {`FD, G2, G3, G4`}.
Neural net structure for all experiments is:
Band probability estimators net structure: 101–100–21 (12100 weights)
Merger probability estimators net structure: 273–300–21 (88200 weights)

The recognition results for MTRAP systems are shown in Tab 7.10. The first two lines show the performance of systems, where CRBS was modified by frequency differentiating operators – one dimensional operator FD and two dimensional operator G2. These systems do not do a good job in

noisy conditions neither, as can be seen from the table, but we expect they will help in combination with the TRAP baseline.

The diagonal modifying operators G3 and G4 achieve slight improvement in clean test set but degrade in case of noise compared to TRAP baseline. This degradation is slightly higher than for plain three band system (see Tab 7.9).

Experiments with dimensionality reduction of MTRAP were also done. The experiment names are `AUR_MO_mt101_DCT51_mvn` for dimensionality reduction using DCT basis and `AUR_MO_mt101_PCA51_mvn` when PCA bases are used for dimensionality reduction. The `MO` is one of the following modifying operators: {`FD, G2, G3, G4`}.
Neural net structure for these experiments is:
Band probability estimators net structure: 50–100–21 (7200 weights)
Merger probability estimators net structure: 273–300–21 (88200 weights)

Results for these experiments are given in Tab. 7.11 and Tab. 7.12. The results are very similar to the ones without dimensionality reduction and alternate without clear tendencies. Generally it is possible to say, that systems with PCA dimensionality reduction are slightly poorer than DCT ones.

7.2.7 Combinations of TRAP systems

These techniques combine informations derived from different TRAP-based systems. Three techniques, which combine the information in different stages of TRAP feature extraction are described in Chapter 6.

Here, the *multi-stream combination* and *vector concatenation* approaches are tested. These combination techniques are easy to implement and improved the final performance in previous experiments. Systems used for combination are TRAP baseline with either FD MTRAP or G2 MTRAP system. Systems with DCT dimensionality reduction are tested too. The performances of individual systems which are going to be combined are given in Tab. 7.13. Note that the baseline `AUR_base_trap101_mvn` system has the best performance for strong noises. This system is outperformed for noises with SNR above 5 dB by adding DCT dimensionality reduction. Other systems have poorer performance.

noise	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
`AUR_base_trap101_mvn`	1.9	6.5	10.7	19.2	37.8	69.0	87.7	33.2
`AUR_FD_mt101_mvn`	2.1	7.6	15.2	33.1	63.1	84.5	90.8	42.3
`AUR_G2_mt101_mvn`	2.3	7.9	14.2	28.7	54.0	78.9	89.6	39.4
`AUR_base_trap101_DCT50_mvn`	1.6	5.6	9.5	18.4	42.0	75.9	89.8	34.7
`AUR_FD_mt101_DCT50_mvn`	2.0	8.4	15.7	32.1	59.4	81.9	89.6	41.3
`AUR_G2_mt101_DCT50_mvn`	2.0	8.3	15.8	32.2	59.7	82.5	90.1	41.5

Table 7.13: WER [%] of further combined systems

7.2.7.1 Multi-stream combination

The multi-stream combination technique is described in section 6.1. All three proposed methods – averaging, logarithmic averaging and inverse entropy weighting – which were tested on Stories-Numbers setup are tested also on Aurora setup.

The results for systems without dimensionality reduction can be seen in Tab. 7.14, with DCT reduction in Tab. 7.15. We can see that the combination of base TRAP and G2 MTRAP systems brings better results than the combination with FD MTRAP. Better performance of G2 modifying operator can be explained by the time averaging attribute of the operator which smooths frame-by-frame differences of critical band energy trajectory and thus also smooths out the differences caused by noise. The best performing combination method is inverse entropy weighting.

AUR_base_ _trap101_mvn +	combi-nation	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
AUR_FD_ _mt101_mvn	lin ave	1.5	4.4	8.9	21.7	50.7	80.5	89.8	36.8
	log ave	1.3	3.8	8.2	20.6	49.4	80.4	89.6	36.2
	inv ent	1.4	3.9	7.9	18.6	44.6	78.0	89.6	34.9
AUR_G2_ _mt101_mvn	lin ave	1.6	4.4	8.5	19.7	45.0	76.5	89.1	35.0
	log ave	1.3	3.9	8.0	18.9	44.7	77.2	89.0	34.7
	inv ent	1.4	3.9	7.5	17.2	41.1	75.5	89.1	33.7

Table 7.14: WER [%] of multistream combination of TRAP based systems without dimensionality reduction

AUR_base_ trap101_DCT50_ _mvn +	combi-nation	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
AUR_FD_ _mt101_DCT50_ _mvn	lin ave	1.3	4.1	8.2	20.4	49.0	79.4	89.1	35.9
	log ave	1.2	3.8	8.0	20.2	49.8	81.2	89.9	36.3
	inv ent	1.3	3.8	7.8	19.0	46.1	79.2	89.5	35.3
AUR_G2_ _mt101_DCT50_ _mvn	lin ave	1.3	4.4	8.7	21.4	51.0	81.1	89.6	36.8
	log ave	1.1	4.0	8.3	21.1	50.9	81.8	90.2	36.8
	inv ent	1.3	3.9	7.9	19.3	47.1	80.4	90.3	35.8

Table 7.15: WER [%] of multistream combination of TRAP based systems with DCT dimensionality reduction

Multi-stream combination technique helps in cases of SNR bigger or equal to 10dB. The error is almost half of the baseline error in case of weak noise with SNR 20. For strong noises, the resulting system makes more errors than the better one of individual systems. There is no further option to tune the performance for averaging methods except of giving weights to combined streams. This would lead to inclination towards performance of the preferred stream. On the other hand, there is a value which can be tuned in inverse entropy weighting method. This value is the *static threshold* established in Eq. 6.6. The entropy values for one sentence with SNRs 20dB, 10dB and 0dB are shown on Fig 7.1. The entropies of base TRAP system and G2 MTRAP system without dimensionality reduction are shown. The horizontal dotted lines are drawn for entropy values of 1, 1.5, 2, 2.5 and 3.

If the entropy of both systems is below given threshold, the output is combined according to entropy combination. If entropy of both systems is above the threshold, both systems get the same weight which leads to averaging combination. The advantage of thresholding takes place only when one system is below, and the second above the threshold value. Fig. 7.1 shows, that with raising level of noise in signal, the entropies are also raising, and when the threshold is set to one, the entropies are getting into an area above the threshold. The preferred combination is then the averaging combination most of the time.

Table 7.16 shows results for inverse entropy based multi-stream combination of base TRAP system and MTRAP systems with different thresholds. Combined systems do not use dimensionality reduction. We can see that there is certain range of threshold values where the WER stays the same for weak noises. This range decreases with decreasing SNR and cannot be observed for SNR 0 and smaller (may be also due to large step in threshold values). The WER has its minimum in this range and WER increases on the sides. The threshold which gives the best overall performance gives the

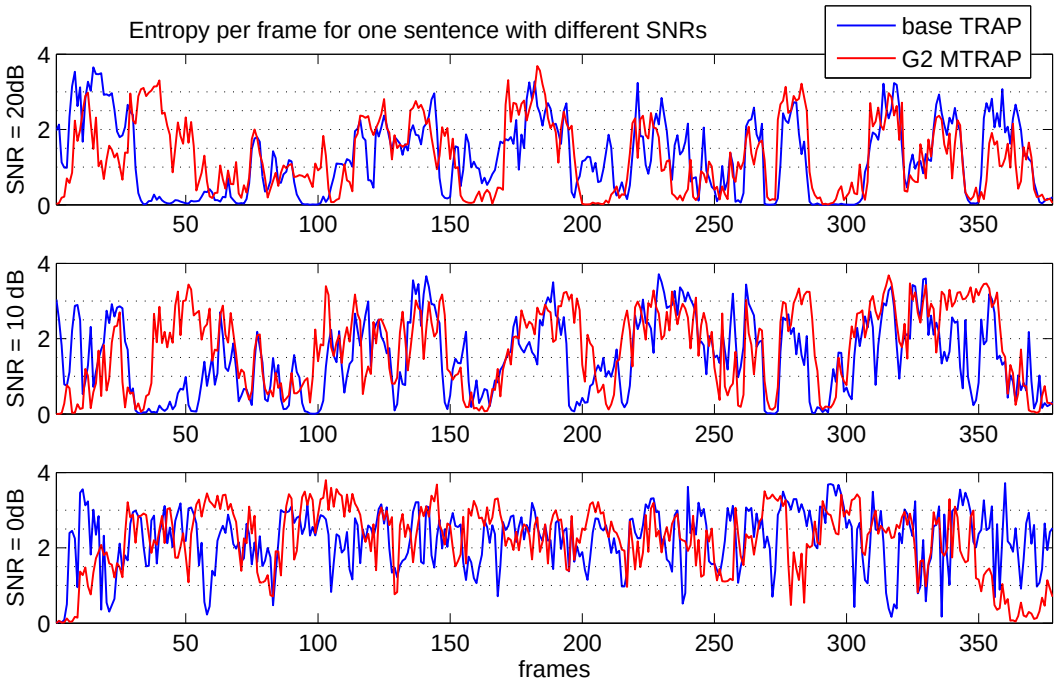


Figure 7.1: Entropy for sentence FFN_4880701A with different SNRs

best performance also for individual noise condition – this means that the threshold does not have to be tuned for individual SNR but can be set to the overall optimal value. With this optimal threshold, better performance is achieved than with threshold initially set. It is also the first system, which has global average WER lower than the TRAP baseline system.

7.2.7.2 Vector concatenation

The vector concatenation technique is described in section 6.3. This technique is simple and very good results were obtained on Stories-Numbers experimental setup.

System AUR_VC_BT_MO concatenates base TRAP and MTRAP vector on the input of band probability estimator. The MO is one of {FD, G2} modifying operators.
Neural net structure for these experiments is:
Band probability estimators net structure: 202–100–21 (66900 weights)
Merger probability estimators net structure: 315–300–21 (100800 weights)

System AUR_VC_DCT50_BT_MO concatenates base TRAP and MTRAP vector after DCT dimensionality reduction on the input of band probability estimator.
Neural net structure for these experiments is:
Band probability estimators net structure: 100–100–21 (36300 weights)
Merger probability estimators net structure: 315–300–21 (100800 weights)

The results are shown in Tab. 7.17. This concatenation technique brings further improvement in performance over the multi-stream combination. Its advantage is evident namely for signals with strong noises (SNR smaller than 10dB). For weak noises, the results are similar to the one obtained by inverse entropy multi-stream combination. The best system is AUR_VC_BT_G2.

The system takes advantage from different information presented on the input of the band conditioned classifier. Firstly, it sees the time evolution of critical band energy. Also, the energy slope over three bands is encoded by modifying operator for each frame. Further, the time averaging in G2 operator filters out the sudden changes and thus it is making the values more stable and noise

AUR_base_ _trap101_mvn +	thres-hold	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
AUR_FD_ _mt101_mvn	1.0	1.4	3.9	7.9	18.6	44.6	78.0	89.6	34.9
	1.5	1.5	3.7	7.4	17.1	41.3	76.2	89.1	33.8
	2.0	1.4	3.6	7.0	16.2	39.4	74.8	89.0	33.1
	2.5	1.3	3.6	7.0	16.3	38.2	72.4	87.9	**32.4**
	3.0	1.3	3.9	7.8	18.3	43.5	76.5	88.9	34.3
AUR_G2_ _mt101_mvn	1.0	1.4	3.9	7.5	17.2	41.1	75.5	89.1	33.7
	1.5	1.4	3.7	7.1	16.1	38.4	72.9	88.5	32.6
	2.0	1.4	3.7	7.0	15.7	37.1	71.2	88.4	32.1
	2.5	1.5	3.7	7.1	15.5	37.0	71.5	88.0	**32.0**
	3.0	1.5	3.9	7.5	16.9	40.3	74.2	88.5	33.3

Table 7.16: Inverse entropy weighting combination with different thresholds of systems without dimensionality reduction

noise	clean	SNR20	SNR15	SNR10	SNR5	SNR0	SNR-5	average
AUR_VC_BT_FD	1.6	4.2	7.6	15.6	36.4	70.9	88.5	32.1
AUR_VC_BT_G2	1.3	3.9	6.9	13.9	33.5	68.1	87.9	**30.8**
AUR_VC_DCT50_BT_FD	1.4	4.0	7.3	16.1	39.8	74.6	89.0	33.2
AUR_VC_DCT50_BT_G2	1.4	4.0	6.9	15.5	37.8	73.3	88.4	32.5

Table 7.17: WER [%] of vector concatenation combination

robust. If classifier trained on such representation sees a new data on its input, both characteristics – time evolution and frequency slope of the energy – contribute to the classification. If there are more aspects characterizing the specific class, it is less likely that noise will change all of them and that the resulting vector will be classified as different class. Such a random change will rather cause unsure output of the band-conditioned classifier. This is actually better situation than to be sure about a wrong output. Also, if the noise corrupts only one characteristic, the correct decision can still be made base on the other one.

7.3 Summary

This chapter describes the experimental setup for noisy speech recognition. The performance of different TRAP-based systems under noisy conditions is examined. The performances of standard MFCC features and basic TRAP features (AUR_base_trap101_mvn) are evaluated. In clean case, the MFCC features perform much better than the basic TRAP features. When noise is present, the TRAP features are superior to MFCC features. For the highest noise level – SNR -5 dB – both features reach the "one out of N" word error rate of about 90%[2]. The performance of these baselines is depicted in Fig. 7.2.

Note, that the y-axis with WER values is logarithmic to preserve relevant information about differences between two systems. The 0.5% difference is important when WERs are around 1% and minor for results around 60% WER. The *Inf* on x-axis denotes clean speech, which has 0 noise energy – infinite signal to noise ratio.

[2]The "one out of N" word error rate is 90.9% assuming that all words occurs equally.

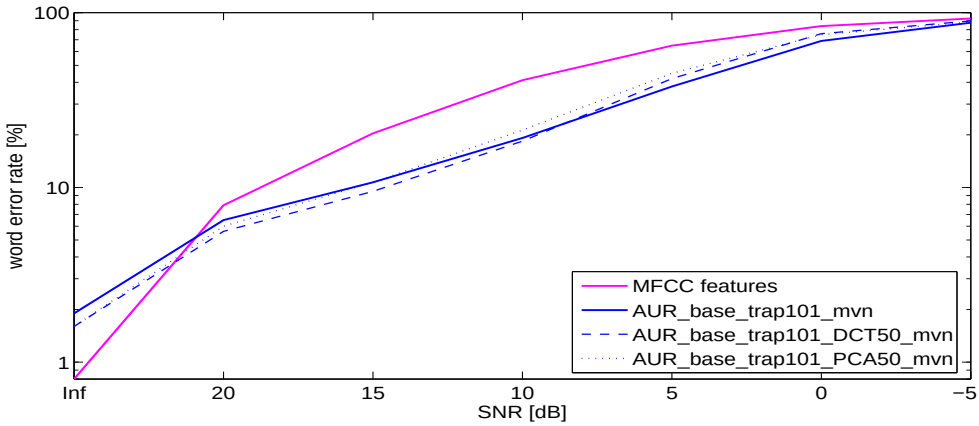


Figure 7.2: WER [%] for MFCC features, TRAP baseline and TRAP with dimensionality reduction

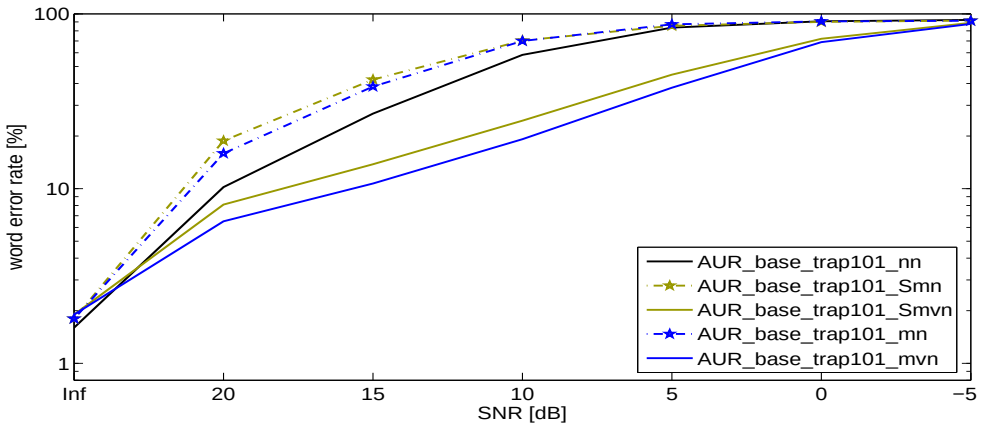


Figure 7.3: WER [%] for basic TRAP system with different normalization schemes

The effect of normalization scheme on TRAP system performance was examined first. The results in Fig. 7.3 show, that the robustness of TRAP system to real-word, full-spectrum additive noise is limited. The simple mean and variance normalization technique [59] brings significant improvement in recognition of noisy speech using TRAP-based features. This is due to the effect of noise on the critical band spectrogram. The noise increases the energy level, which causes decrease of dynamics of temporal trajectory and changes of its shape. When mean normalization is done, the results are even worse, as the mean subtraction leads to the loss of the level of the band trajectory causing resulting signal to oscillate around zero. With increasing noise level, the mean normalized trajectories are closer to zero. Only when dynamics is preserved by mean and variance normalization, the system is more noise robust. By changing the variance, there is a small degradation in performance compared to the system with no normalization for clean data, but big improvement in noisy conditions.

TRAP systems where dimensionality reduction is applied on the critical band trajectory were examined further. Both, DCT (`AUR_base_trap101_DCT50_mvn`) and PCA (`AUR_base_trap101_PCA50_mvn`) dimensionality reduction techniques were tested. Word error rate as a function of SNR for TRAP with dimensionality reduction is shown in Fig. 7.2. Applying dimensionality reduction improves TRAP baseline performance on clean set and weak noise sets with SNR $>$ 10 dB. The performances decrease below the TRAP baseline performances for SNR $<$ 10 dB.

The TRAP system which concatenates three adjacent bands `AUR_3b_trap101_mvn` also improves over TRAP baseline on clean set and sets with SNR $>$ 10 dB as can be seen in Fig. 7.4. But

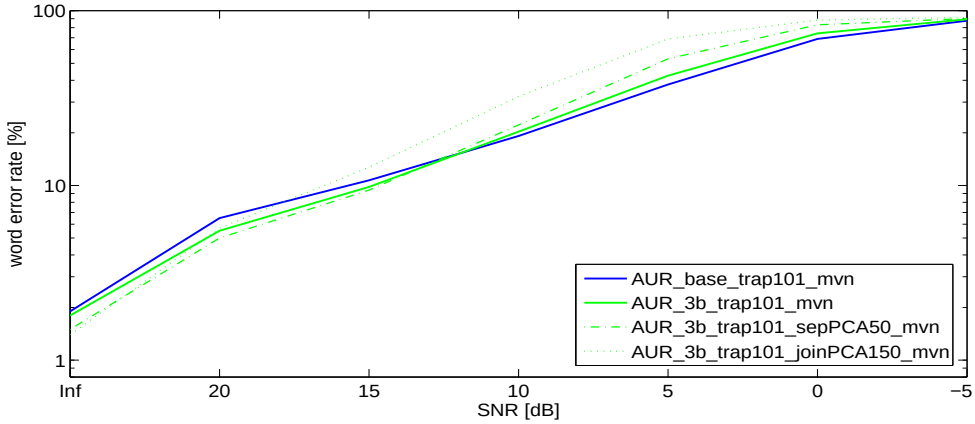


Figure 7.4: WER [%] for TRAP baseline and 3band TRAP systems without and with PCA dimensionality reductions

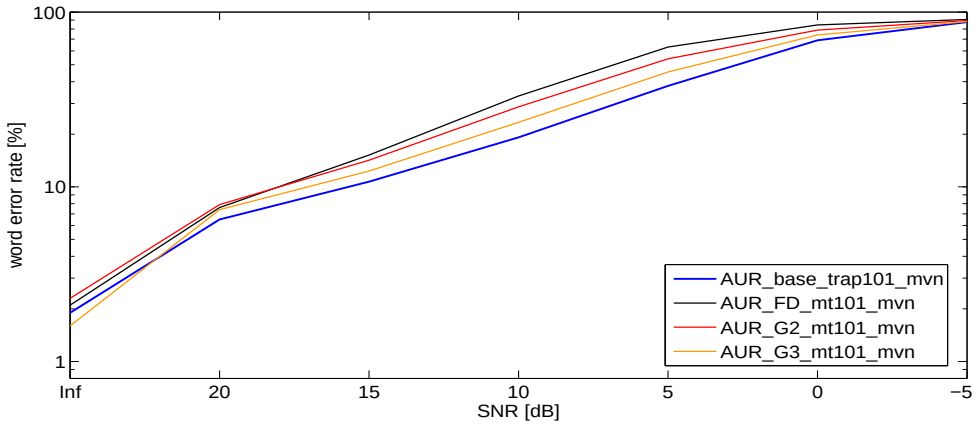


Figure 7.5: WER [%] MTRAP systems without dimensionality reductions

also here, the degradation under noises stronger than 10 dB occurs. When the dimensionality reduction is applied, further improvement for clean set is observed, but the degradation in noisy case increases. The crossing point with TRAP baseline is around SNR = 15 dB. Systems which do the dimensionality reduction first and concatenate its results (`AUR_3b_trap101_sepDCT50_mvn`, `AUR_3b_trap101_sepPCA50_mvn`) show smaller sensitivity to noise compared to the ones which concatenate TRAP vectors first and then do the dimensionality reduction (`AUR_3b_trap101_joinDCT150_mvn`, `AUR_3b_trap101_joinPCA150_mvn`). Our explanation is that noise in one critical band affects all points in the resulting vector, whereas when the dimensionality reduction is done first, the noise in one critical band affects only its part of the resulting vector. The most noise sensitive system is the one using PCA bases on concatenated TRAP vectors for dimensionality reduction. The results for TRAP baseline and TRAP systems which concatenate three adjacent bands and do the PCA dimensionality reduction are shown in Fig. 7.4.

The results for MTRAP systems without dimensionality reduction are shown in Fig. 7.5. We can see that there is some variation around TRAP baseline performance for clean set, but for noisy sets, no modification performs better than the TRAP baseline. As the results are almost the same for G3 and G4 MTRAP, only the G3 results are shown. The dimensionality reduction has only slight effect for MTRAP systems – the results alter without clear tendencies. The PCA dimensionality reduction seems to be more vulnerable to noise than the DCT.

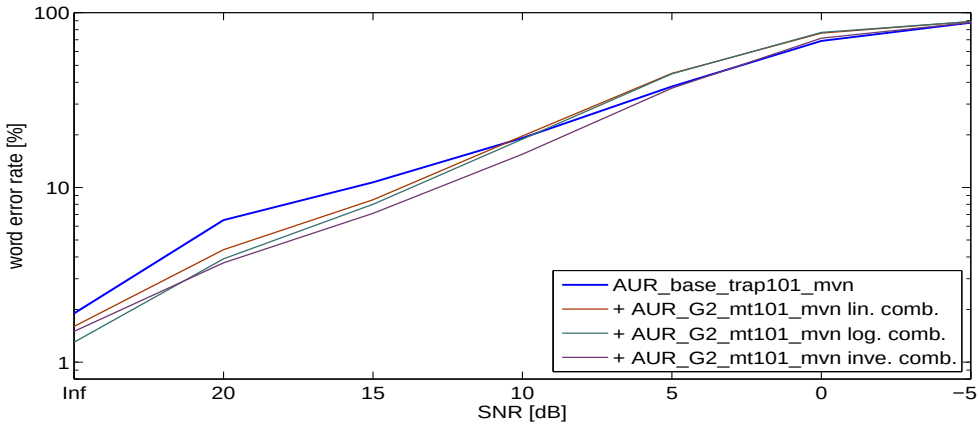


Figure 7.6: WER [%] for multistream combination of TRAP baseline and G2 MTRAP without dimensionality reductions

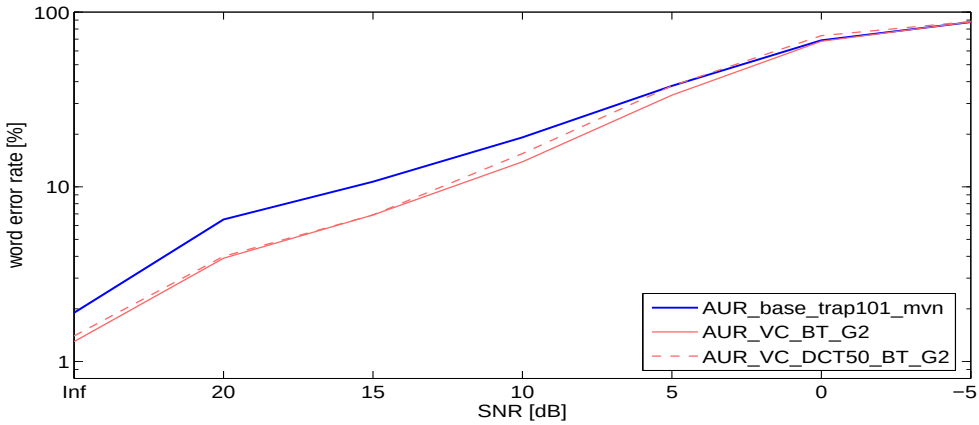


Figure 7.7: WER [%] for vector concatenation combination of TRAP baseline and G2 MTRAP without and with dimensionality reductions

The combinations of two systems were also tested. We combine TRAP baseline with FD or G2 MTRAP using multi-stream approach with different combination techniques – linear averaging (lin. comb.), logarithmic averaging (log. comb.) and inverse entropy combination (inve. comb.). All methods perform well in clean test set, achieving better performance than any of the combined systems. When tested on the sets with SNR $>$ 10 dB, the combination outperforms all other TRAP-based systems. For test sets with SNR $<$ 10 dB, the performance is poorer than the performance of TRAP baseline, but the difference between the systems is small. The inverse entropy combination technique with optimal threshold (threshold = 2.5) gives the best multi-stream combination performance and for low SNRs, it performs about the same as TRAP baseline. The results for multi-stream combination of TRAP baseline with G2 MTRAP without dimensionality reduction are shown in Fig. 7.6. Slightly better performance on clean test set can be achieved by combination of system which uses the dimensionality reduction.

The vector concatenation combination approach leads to further improvement for noisy test sets. This technique has better performance than the TRAP baseline and meets it only for SNR worse than 5 dB. The best performing vector concatenation – `AUR_VC_BT_G2` – reduces the WER over all test sets. The results for vector concatenation combination of TRAP baseline with G2 MTRAP without and with dimensionality reduction are shown in Fig. 7.7.

Other observation from experiments with combination of TRAP systems is better performance of combination of baseline TRAP system with G2 MTRAP system than with FD MTRAP system. We explain this behavior by the time averaging property of G2 modifying operator which partly smooths out the frame-by-frame changes caused by the noise and thus enhance noise robustness of the system.

To summarize, it can be stated, that dimensionality reduction can reduce the error rate for SNR > 10 dB over the system without dimensionality reduction in exchange for degradation of performance for test sets with SNR < 10 dB. For the SNR $= -5$ dB, all systems reach the same "one out of N" performance of about 90% WER.

The robust speech recognition is a field which started to be explored in late seventies in [9]. Authors proposed noise suppression by subtraction the mean of noise distribution from current feature vector. The technique is dependent on proper speech-nonspeech detection and correct distribution estimation from limited data. In almost four decades, various techniques have been developed in order to decrease the impact of noise on ASR systems, such as Wiener filtering and spectral subtraction [7, 58], noise robust features – J RASTA filtering [37], model decomposition and adaptation [89, 55] and missing data compensation [17]. These techniques used with cepstral features could be used also with TRAP-based features, but testing them is beyond the scope of this thesis.

Chapter 8

TRAP features for LVCSR

The most difficult task in speech recognition is Large Vocabulary Continuous Speech Recognition (LVCSR). Besides acoustic modeling, language modeling and lattice rescoring – each of them research topic on its own – play important role in LVCSR. Due to this large processing, it is much harder to say, what role play features on the beginning of the speech recognition process. They can bring improvement in acoustic decoding, but this can be canceled by the further processing. Since in our experiments this processing is fixed for evaluation of different features, the change in final word error rate (WER) can be ascribed to the quality of features.

We examined the performance of TRAP-based probabilistic features in two LVCSR tasks. First, the probabilistic features are tested on recognition of meeting speech, where we tried to find the optimal features for this task, which are then combined with standard cepstral MFCC features. In the second task, TRAP-based probability estimates are used for recognition of conversation telephone speech in system where they are combined with probabilities obtained from different source first and then converted to probabilistic features and combined with PLP cepstral features.

8.1 Meeting speech recognition

Meeting speech recognition (automatic transcription of meetings) is extremely difficult task bringing new problems into LVCSR such as overlapping speech, non-native speakers and vocal and environmental noises. Automatic meetings transcription and other fields of interest associated with meeting audio-video data have been investigated in three large European Union projects: Multi-Modal Meeting manager[1] (M4) and its follow-ups: Augmented Multi-party Interaction[2] (AMI) and Augmented Multi-party Interaction with Distant Access[3] (AMIDA).

8.1.1 Used data

We use the ICSI meetings database [46] for experiments. It contains recordings of real spontaneous meetings with cross-talks, unfinished words, background speech and all kinds of speaker noises. The speakers in the database are often non-native US English speakers. All these conditions make the recognition very difficult. Two sets of data were defined:

Training set consists of 39.4 hours of speech from 40 meetings. There are 26 speakers in the set, 4 females and 22 males of which 12 are native and 14 non-native.

Test set consists of 1 hour of speech randomly selected from three meetings. There are 7 speakers – 2 females and 5 males of which 4 are native and 3 non-native.

[1] www.m4project.org
[2] www.amiproject.org
[3] www.amidaproject.org

label	index	frames	perc%	label	index	frames	perc%
aa	0	32825	0.90	m	23	79230	2.18
ae	1	105134	2.89	n	24	144664	3.98
ah	2	57341	1.57	ng	25	44420	1.22
ao	3	36019	0.99	ow	26	78188	2.15
aw	4	19430	0.53	oy	27	5918	0.16
ax	5	131185	3.61	p	28	41232	1.13
ay	6	83110	2.28	r	29	80569	2.21
b	7	35658	0.98	s	30	175021	4.81
ch	8	22395	0.61	sh	31	23229	0.63
d	9	56765	1.56	t	32	135916	3.74
dh	10	69339	1.90	th	33	22975	0.63
dx	11	11683	0.32	uh	34	10690	0.29
eh	12	58108	1.59	uw	35	51174	1.40
er	13	54342	1.49	v	36	34015	0.93
ey	14	57704	1.58	w	37	73704	2.02
f	15	50366	1.38	y	38	47828	1.31
g	16	19334	0.53	z	39	88469	2.43
hh	17	22632	0.62	puh	40	91981	2.53
ih	18	66969	1.84	pum	41	27110	0.74
iy	19	108108	2.97	sil	42	817753	22.50
jh	20	14671	0.40	voc	43	225322	6.20
k	21	84391	2.32	hes	44	60899	1.67
l	22	75389	2.07	total		3633205	

Table 8.1: Phoneme coverage in Meeting NN training data.

The data sets were defined so that no speaker occurs in both sets. Only parts of signals with speech activity were taken based on meetings transcriptions. Both sets contain 10282 distinct words of which 113 appear only in the test set. There is no development set for this task and all parameter tuning was done directly on the test set.

8.1.2 Neural net training

The input critical band energies (CRBE) are calculated from speech signal divided into 25 ms frames with 10 ms shift. Each frame is weighted by a Hamming window and FFT is computed. The spectral values are filtered into 19 Bark scaled trapezoidal filters and logarithm is taken.

A 10 hour subset of HMM training data is used for training the neural nets. The data were labeled by forced alignment obtained while training the system (Sec. 8.1.4) with baseline MFCC features (Sec. 8.1.5). The phoneme set consists of 45 phonemes including silence (sil), vocal noise (voc), hesitation noise (hes) and two filler phonemes (puh, pum). The coverage of phonemes in training data is shown in Tab. 8.1. These phonemes were used as targets during neural net training.

8.1.3 Tuning TRAP-based features

We were looking for the TRAP-based features, which gives the best performance on this task. Besides various kinds of TRAP processing, discussed largely in Chapters 4, 5 and 6, we also considered the normalization scheme and size of the temporal context of TRAP vector. Though we did not test all TRAP processing schemes introduced earlier, the search space was quite large considering computation requirements of this task.

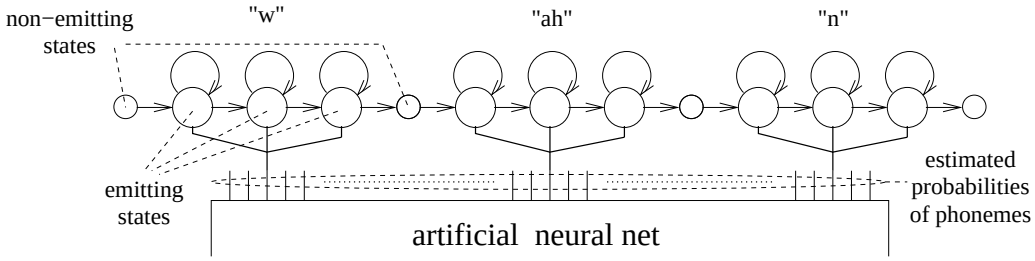


Figure 8.1: Hybrid system scheme.

Based on the previous results on Stories-Numbers experiment we decided to test the *base TRAP features* (Sec. 4.1), *base TRAP features with DCT dimensionality reduction* (Sec 4.2) and their vector concatenation with MTRAP obtained by modifying operators FD and G2 (Chapter 5). The vector concatenation technique is described in Sec. 6.3. The normalization kinds were *no normalization*, *sentence based mean normalization* and *TRAP (vector) based mean normalization*. The time span was examined in range from 21 to 101 frames with step of 20 frames. All these conditions gave us 90 points in search space and it was not feasible to run full system for all of them.

Instead, a hybrid [10, 65] system was designed for fast testing of TRAP based features. Hybrid system does not treat the posteriors as features that are then modeled by mixtures of Gaussians (GMM), but neural net outputs are used directly as emission probabilities. Each emitting state of phoneme models is connected with one neural net output. Since the NN outputs are phonemes, the models based on this NNs can be only context independent phoneme models with the same emission likelihood for all states in one phoneme model. The number of emitting states in one model is the minimum duration of the phoneme. See Fig. 8.1 for scheme of hybrid system where Markov chain for word "one" is depicted and connections from neural net are shown.

We assume, that results obtained from hybrid recognition system will have similar tendencies to the final GMM system. Although both modes, hybrid and GMM, obtain emission likelihood in different way, the decoding process is the same and features for GMM-based recognizer will be derived from acoustic observations used in hybrid system. This makes us believe that the optimization will lead to proper TRAP-based features.

The `ICSI Sprach Core` large vocabulary decoder NOWAY [19] was used. The phoneme models had three emitting states and equal transition probabilities between states. The phoneme set consisted of 45 phonemes shown in Tab. 8.1. The prior probabilities of phonemes were set to be equal.

Used trigram language model was trained on transcriptions of training part of ICSI meetings database (53099 sentences) and whole Switchboard database (248581 sentences). Each sentence from ICSI database was put into language model training corpus five times for better balancing of training data amounts.

The dictionary had 36000 entries including multiple pronunciations of the same word and was created by merging ICSI meetings dictionary with Switchboard dictionary.

All TRAP-based systems were trained with 100 hidden units in frequency conditioned probability estimators and 300 hidden units in the merger. Number of inputs to the frequency conditioned estimators vary from 21 to 101 for the base TRAP or MTRAP systems depending on tested time context. DCT dimensionality reduction returned half of the input points. The DCT bases do not contain the 0th basis – DC offset. When vector concatenation was used, the input size doubled. The input size of merger probability estimator was 765 (17×45) in case of MTRAP and 855 (19×45) otherwise.

The results obtained with hybrid system during TRAP feature optimization are shown in Fig. 8.2. The results for base TRAP and vector concatenation of BD MTRAP and base TRAP with DCT dimensionality reduction only are shown. Over all experiments we see that:

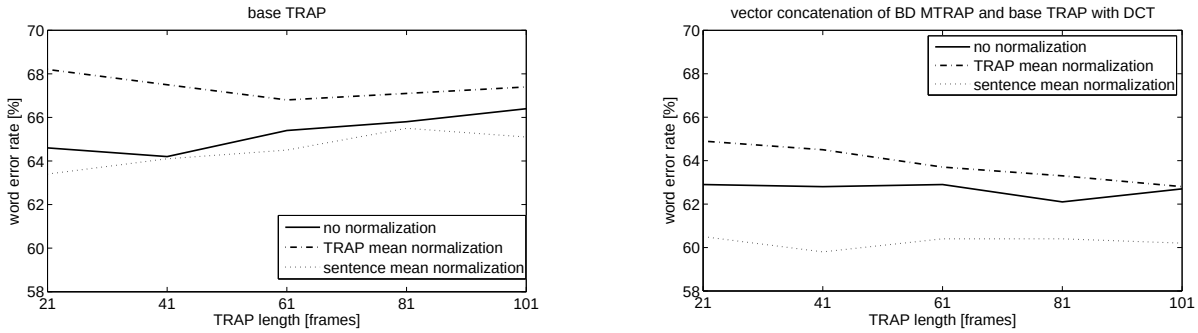


Figure 8.2: Selected TRAP feature optimization (hybrid) results

- TRAP-based mean normalization does not bring any improvement over the system with no normalization.
- The sentence mean normalization brings consistent improvement in WER.
- Optimal TRAP length for these experiments is between 41 and 61 frames.
- Best results – 59.8% WER – were obtained by system with vector concatenation of BD MTRAP and base TRAP with DCT dimensionality reduction.

We further proceeded with the best performing system. The number of hidden units in merger was increased to 1500 and the net was retrained. The hybrid recognition WER decreased to 57.4%. The class posterior estimates were logarithmized and decorrelated using PCA to make them suitable for GMM-HMM recognizer. These features are further denoted as TRAP-opt.

8.1.4 GMM-HMM recognition system

Training of the full recognition system was based on HTK toolkit [97].

Phoneme models have 3 emitting states and they are simple left-to-right models. The silence and noises are also modeled with 3 emitting states and they contain the loop from the last emitting state to the first one. The short pause (sp) model has one emitting state and a loop from the entry state to the exit state.

The training of the GMM models was done from flat start in the following steps:

- Context independent phoneme models with one Gaussian mixture component were initialized by the global mean and variance and reestimated with the help of word level transcriptions.
- These models were used for initialization of context dependent models. Triphones occurring in training data were created and reestimated.
- HMM states were tied using decision tree resulting in 3000 to 3500 tied states. The tying was different for each feature kind – optimal for given features.
- The output 8 Gaussian mixture triphone models were generated by splitting mixtures and trained by iterative Baum-Welch reestimations.

We used the same language model and dictionary for recognition with HMM system as was used for hybrid system (see Sec. 8.1.3). The decoding was done in the following steps:

- Reduction of possible hypotheses by fast stack decoder HDUcode [95]. Word internal (WI) triphones and trigram language model were used to generate a set of lattices. According to our preliminary results, the recognition accuracy of the Viterbi search was slightly higher but stack

features	WI models	WE models	speaker adaptation
MFCC	53.6	48.0	47.5
TRAP-opt	53.6	50.7	50.6
(MFCC + TRAP-opt) PCA 40	52.8	49.1	48.6
(MFCC + TRAP-opt) HLDA 39	—[a]	45.8	**45.4**

[a]HLDA starts with already trained models, therefore this part cannot be evaluated

Table 8.2: WER [%] for MFCC and TRAP-opt features and their combinations

decoder was significantly less time and memory consuming. Strong pruning was used for further acceleration with loosing 2% accuracy only. This decreased time needed for decoding from 150× to 8× real time.

- Acoustic rescoring by a time synchronous decoder HVite [97] and cross-word (word external WE) context dependent triphones.

- Maximum likelihood linear regression (MLLR) speaker adaptation of acoustic models. This process was performed iteratively using recognition results from previous iteration. It was done for each speaker separately. This approach does not increase the demand for computation power significantly because only models in the rescoring (faster) part of the system were adapted.

8.1.5 TRAP and MFCC features and their combination

The baseline features were MFCC with twelve cepstral features appended with zeroth cepstral coefficient computed for 25 ms of speech every 10 ms. The first and second derivatives were added to the direct features. The cepstral mean and variance normalization was performed over the utterance.

Models trained with baseline MFCC features are used to generate phoneme forced alignment for neural net training.

TRAP based – TRAP-opt – features are obtained as described in section 8.1.3.

Results from different stages of the recognition are shown in the first part of Tab. 8.2. We see that both MFCC and TRAP-opt features have the same performance for word internal context dependent models. Introduction of more complex word external models further improves the performance for both features, however the improvement for TRAP-opt features is smaller than for MFCCs. Speaker adaptation brings further improvement, but again this is much smaller in case of TRAP-opt features.

This behavior of TRAP-opt features can be explained by the way they are obtained. The classifiers in TRAP processing were trained to discriminate between context independent classes which consequently leads to reduction in variability within one class. Thus the differences in context dependent phoneme classes and speaker variability are minimized. Therefore we see only small improvement when more complex HMM models are used.

We further tested the combination of MFCC and TRAP-opt features as proposed in [6]. The MFCC coefficients with their derivatives were first normalized over all data and then the TRAP-opt features (these are normalized by the eigen values of their PCA) were appended to them. PCA was performed to reduce the dimensionality of the MFCC + TRAP-opt feature vector to 40, 50 and 60 coefficients. The first part of recognition with word internal triphones was performed for these sizes of feature vector. Minimum WER – 52.8% – was found for feature vector with 40 coefficients. The full recognition system was evaluated only with these features.

From the results shown in the second half of Tab. 8.2, we see that the improvement obtained for WI models did not carry through all the system. Moreover, the combined features did not achieve the performance of MFCC features alone.

To see whether the poorer performance is due to the TRAP based features or whether it is due to the improperly designed combination, the combination using Heteroscedastic Linear Discriminant Analysis (HLDA) [54] was tested. HLDA relaxes the LDA assumption of equal class conditional covariance matrices and allows to derive such projection that best decorrelates features associated with each particular class. The Gaussians in GMM models are considered as HLDA classes. When performing the dimensionality reduction, HLDA allows to preserve useful dimensions, in which classes are best separated.

The results are shown on last line of Tab. 8.2. We see that combination of MFCC and TRAP based features can bring improvement over MFCC baseline features also in complex GMM-HMM recognition system when proper feature combination is designed.

8.2 Conversation Telephone Speech (CTS) recognition

The recognition of conversation telephone speech is another challenging task in these days. The recognizer has to deal with variety of transmission channels, background noises, informal and slang speech, crosstalks and other phenomena of casual speech.

The experimental setup used in the following experiments was developed at ICSI, Berkeley.

8.2.1 Used data

The training set used for both Neural net and GMM-HMM training consists of about 68 hours of conversational speech data from three sources: English CallHome, Switchboard I with transcriptions from Mississippi State University, and Switchboard Cellular. This training set corresponds to the one used in [67] without Switchboard Credit Card data. Training of both MLPs and GMM-HMMs was done separately for each gender.

The 2001 Hub-5 development data were used to tune recognizer parameters. The parameters were optimized to maximize performance on this set and then used for the recognition of evaluation set.

The reported results are obtained on 2001 Hub-5 evaluation data, a large vocabulary conversation telephone speech test set consisting of 6.27 hours of speech with 62 890 words.

8.2.2 Neural net training

The inputs are logarithmic critical band energies mean and variance normalized over a conversation side. The critical band energies are derived from speech in a standard way of feature computation, i.e. splitting the speech signal into 25 ms frames with 10 ms shift, applying a Hamming window, computing FFT and integrating the energies into critical band filter-bank.

The training data were labeled by forced alignment using 46 phonemes including silence (sil), laughter (lau), two filler phonemes (puh, pum) and word fragment interruption point (fip). The coverage of phonemes in training data is shown in Tab. 8.3. These phonemes were used as targets during neural net training. Separate neural nets were trained for each gender.

8.2.3 *"Combined-Augmented"* feature extraction

TRAP-based features in the ICSI CTS experiments are part of more complex feature extraction. Resulting features are called *Combined-Augmented* features. The block diagram of the feature extraction is shown in Fig. 8.3. The explanation of individual blocks follows:

The upper branch of the diagram shows so called "conventional" features. They consists of 12th order PLP coefficients [33] plus energy computed over a 25 ms frame window every 10 ms. 1st, 2nd and 3rd order derivatives are calculated and appended together to yield 52 dimensional feature vector (PLP+Δ+$\Delta\Delta$+$\Delta\Delta\Delta$). The features are mean and variance normalized over a conversation side. The

index	label	male		female	
		frames	perc%	frames	perc%
0	sil	3263229	27.95	3420537	26.55
1	aa	135591	1.16	151461	1.17
2	ae	393653	3.37	448899	3.48
3	ah	171199	1.46	202330	1.57
4	ao	131906	1.13	142672	1.10
5	aw	91619	0.78	101018	0.78
6	ax	503844	4.31	533986	4.14
7	ay	338965	2.90	407684	3.16
8	b	127367	1.09	136880	1.06
9	ch	42575	0.36	50260	0.39
10	d	188988	1.61	213742	1.65
11	dh	184850	1.58	187174	1.45
12	dx	58601	0.50	60916	0.47
13	eh	186977	1.60	213228	1.65
14	er	193526	1.65	206011	1.59
15	ey	184947	1.58	199584	1.54
16	f	133543	1.14	141387	1.09
17	fip	28105	0.24	23142	0.17
18	g	93002	0.79	92851	0.72
19	hh	164013	1.40	226316	1.75
20	ih	215913	1.84	239375	1.85
21	iy	313927	2.68	348476	2.70
22	jh	52806	0.45	55259	0.42
23	k	257123	2.20	284644	2.20
24	l	291882	2.50	319411	2.47
25	lau	198730	1.70	390362	3.03
26	m	203812	1.74	218080	1.69
27	n	428829	3.67	469665	3.64
28	ng	91599	0.78	106204	0.82
29	ow	258056	2.21	380985	2.95
30	oy	12544	0.10	14984	0.11
31	p	140505	1.20	133906	1.03
32	puh	330934	2.83	247337	1.91
33	pum	126749	1.08	190525	1.47
34	r	292286	2.50	309484	2.40
35	s	413025	3.53	434712	3.37
36	sh	61117	0.52	68580	0.53
37	t	412162	3.53	451298	3.50
38	th	66269	0.56	73338	0.56
39	uh	34791	0.29	43868	0.34
40	uw	150771	1.29	173354	1.34
41	v	103084	0.88	108041	0.83
42	w	208205	1.78	243057	1.88
43	y	181434	1.55	205588	1.59
44	z	203637	1.74	208053	1.61
45	zh	4800	0.04	4195	0.03
total		11671490		12882859	

Table 8.3: Phoneme coverage in CTS NN train data

dimensionality of these features is reduced to 39 by HLDA [54] and the resulting vector is denoted HLDA(PLP+3d). These features represent the short-term information.

In the middle of the scheme, we see a classifier whose input are 9 consecutive frames of 12th order PLP coefficients plus energy computed over a 25 ms frame window every 10 ms. The first two derivatives are appended to each feature vector and the feature vectors are normalized to have zero mean and unity variance over a conversation side. The total size of MLP input vector is $(12+1)\times3\times9 = 351$ coefficients. The used classifier is a simple feed-forward three layer perceptron as described in Sec 3.4. Its size is 500 thousand weights. The training targets are 46 phoneme classes used in SRI decoder shown in Tab 8.3. The training target (or the MLP output during the forward pass) is associated with the center frame of the input which has four frames context in future and four in past. These PLP-based posteriors were successfully tested in TANDEM speech recognition [34, 21, 20, 82]. The PLP-based posteriors represent the mid-term information coming from approximately 100 ms of speech.

The lower branch describes the computation of TRAP-based posteriors. First, the power spectrum is computed over a 25 ms frame window every 10 ms. The power spectrum values are then integrated by 15 triangular filters and their outputs are logarithmized. The log-critical band energies are normalized to have zero mean and unity variance over a conversation side. The input to the *temporal classifier* is formed by 51 consecutive frames so its size is $15 \times 51 = 765$. The size of temporal classifier is 500 thousand weights. The structure of temporal classifier is discussed in the next section. The training targets are 46 phoneme classes used in SRI decoder shown in Tab 8.3. Again, the target label or classifier output is associated with the center frame of the input which has 25 frames context in future and 25 in past. The TRAP-based posteriors represent the long-term information covering 500 ms of speech.

Because the MLP trained on different input features can provide different output at given time, combination of multiple phoneme posteriors obtained in different ways can yield much better estimates of phoneme posteriors. Thus the TRAP-based posteriors are combined with PLP-based posteriors estimates in the lower right part of the scheme. The frame-wise multi-stream combination techniques were introduced in section 6.1. Here, the inverse entropy weighting with static threshold [63] was used to combine the TRAP-based and PLP-based phoneme posteriors. The threshold value is set to 1. The resulting posteriors are denoted combined posteriors.

After the multi-stream combination, phoneme posteriors are transformed using the logarithm and decorrelated using PCA. Only 25 most important coefficients are kept after the PCA. This was empirically found to be optimal for this task. Resulting vectors are called *MLP-based features* or *probabilistic features*. The probabilistic features are normalized to have a zero mean and unity variance over a conversation side.

The probabilistic features are appended to the conventional features HLDA(PLP+3d) as proposed in [6], where authors discuss feature performance in noisy conditions. MLP-based features can give poor estimate of phone posteriors in noise. Appending conventional features with features derived using MLP might help the HMM recognizer to come up with correct classification. The final features which are created by the concatenation of HLDA(PLP+3d) and probabilistic features are called *combined-augmented* features. Note, that the feature extraction is gender dependent, so we have male and female version of each classifier.

The place of our interest lies in the area marked by dashed line in Fig. 8.3. There, the log-critical band energies are converted to estimates of class posteriors. We replace this part of the *combined-augmented* feature extraction by our techniques for TRAP-based probability estimation and the rest of it will stay the same. Thus we can evaluate the effect of different TRAP processing on quality of the features through recognition WER.

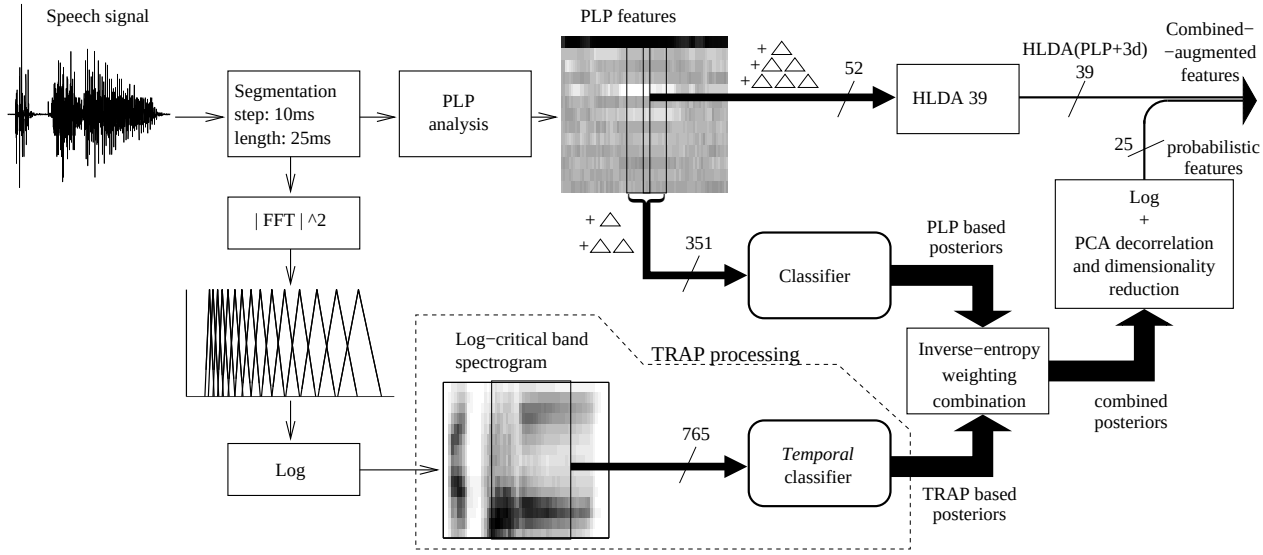


Figure 8.3: The scheme of the ICSI feature extraction

8.2.4 Temporal classifiers

So far, the temporal classifier used in TRAP processing was a two stage classification composed by several three-layer MLPs. In the first stage we estimated band (frequency) conditioned class probabilities. Input of the band-conditioned classifier is a log-energy trajectory from a single critical band. There are as many band-conditioned class probability estimates as critical bands in the analysis – in our case 15. The band-conditioned estimates are merged to give an overall estimate by another classifier: merger. The input for merger classifier is created by concatenation of all band-conditioned class probabilities after a logarithm. The target classes are the same for both stages. The illustration of the two stage temporal classifier is in Fig. 8.4 and we denote this classifier as 2-STAGE.

ICSI researchers B. Chen and Q. Zhu were investigating the 2-STAGE temporal classification approach. They interpreted the first to second layer connections of the band-conditioned classifiers as a kind of learned matched filters useful for phonetic classification. Then they asked two questions:

1. ***Can we skip the mapping from the output of the matched filters to critical band phone posteriors?*** The hypothesis is that the important phonetic information is already captured by the matched filters and the mapping to the phone posteriors may be inaccurate.

2. ***Is there a better way to learn critical band matched filters?*** The idea is to train the matched filters in one step as a part of the overall training procedure.

The research in this area is published in [12, 14, 15] and its outcome were two new architectures of temporal classifier:

- Hidden Activation TRAPS (HATS) are preserving two stage training process. The first stage of training is the same as for 2-STAGE temporal classifier. When the band-conditioned classifiers are trained, the hidden to output layer part is removed, leaving only the outputs ("activations") of the hidden layer – matched filters. The second stage takes as input the concatenation of activation of all band-conditioned matched filters and transforms them to overall phone probability estimates. The scheme of HATS temporal classifier is depicted in Fig. 8.5.

- Tonotopic Multi-Layered Perceptron (TMLP) is a single four-layer MLP which preserves the structure of the TRAP (HATS) temporal classifier. The first hidden layer consists of several groups of hidden units, each of which is constrained to receive the input only from a single

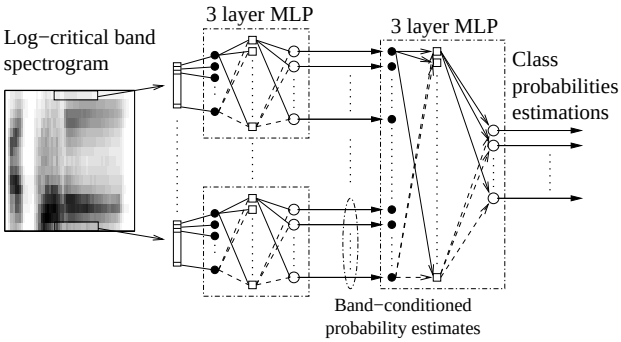


Figure 8.4: 2-STAGE temporal classifier

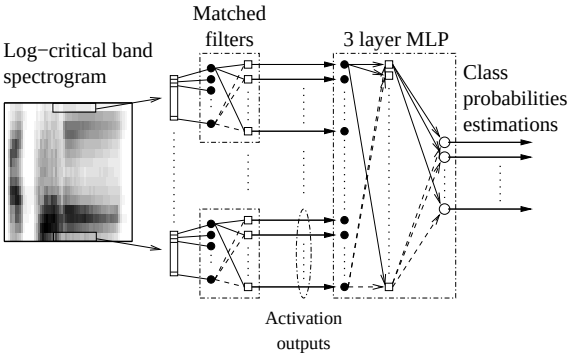


Figure 8.5: HATS temporal classifier

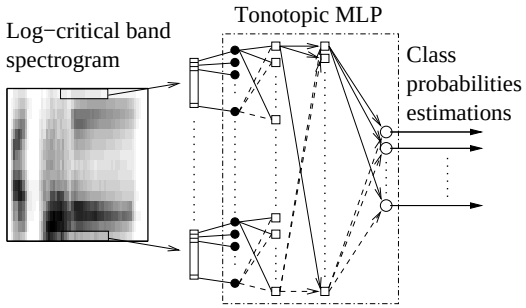


Figure 8.6: Tonotopic MLP temporal classifier

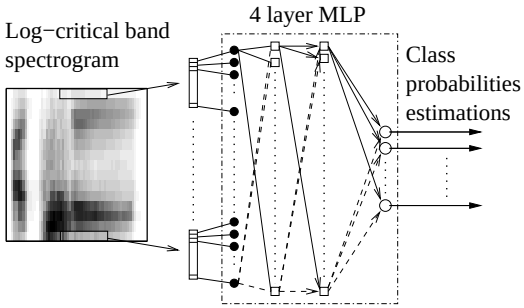


Figure 8.7: Unconstrained 4 layer MLP temporal classifier

critical band. These critical band hidden units correspond to discriminatively trained matched filters in HATS system but they are learned as a part of the overall back-propagation training algorithm. The second hidden layer and the output layer combine the outputs of matched filters across all frequencies to obtain phoneme posterior estimates. The scheme of TMLP temporal classifier is shown in Fig. 8.6.

An unconstrained three or four layer MLP (see Fig. 8.7) can be also used as temporal classifier.

8.2.5 GMM-HMM recognition system

Used GMM-HMM recognition system is based on SRI's DECIPHER [84]. Only the first pass decoding is used in our experiments. It uses gender dependent word-internal (WI) triphone acoustic models and bigram language model.

The pronunciation dictionary is based on the version 0.4 of the CMU pronunciation dictionary. The "multiwords" [24] are added to the dictionary. Multiword is any bigram or trigram occurring more than 200 times in training data and having different pronunciation in continuous speech than the words have themselves or in different context. Multiwords in language model capture frequent higher-order N-grams so it is possible to get local four- or five-grams.

Used language model (LM) was a multiword bigram backoff LM containing about 1.3M bigrams. The LM was trained on Switchboard-1, CallHome and Broadcast News. Separate LMs were created for each corpus and then statistically interpolated into a single backoff model.

The recognizer parameters are tuned on independent tuning set. First, it is important to tune the Gaussian weight which is a scale factor associated with Gaussians in mixtures. This compensates larger dynamic scale of log-likelihood caused by longer feature vector. The effect of this weight is similar to effect of scale factor in HTK. Second, we tune normalizing parameters for N-best rescoring

features	WER [%]
HLDA(PLP+3d)	37.2
PLP-based posterior features	41.2
TRAP-based posterior features	45.9
HLDA(PLP+3d) + PLP-based posterior features	36.1
HLDA(PLP+3d) + TRAP-based posterior features	36.9
"combined-augmented" features	**34.6**

Table 8.4: Results on CTS experiment for each branch of "combined-augmented" feature extraction and their concatenations

to obtain the best WER. These tuning parameters are specific to DECIPHER recognizer system. The efforts of incorporating the probabilistic features into SRI's system are described in [67, 98, 66, 99].

8.2.6 Prior results on CTS task

Before we describe different TRAP processing and show the results, the performance of each feature extraction branch itself is shown. The performance of HLDA(PLP+3d), TRAP-based and PLP based posterior features are given in the upper part of Tab 8.4. The results of features created by concatenation of HLDA(PLP+3d) and one kind of posterior features are given in the second part of Tab. 8.4. The results are adopted from [14]. Note that the TRAP processing done at ICSI differs from what we standardly do - the logarithm is not applied on the output of band-conditioned classifiers. Processing the classifier outputs by logarithm is very similar to replacing the softmax non-linearity by a linear function. That is why we show here results denoted as "TRAPS Before Softmax" in [14]. The results for full "combined-augmented" features with 2-STAGE temporal classifiers are shown in the last part of Tab. 8.4. This is our baseline system.

8.2.7 Experiments with different temporal processing

The complex combined-augmented feature extraction do not require only good classification by the temporal (TRAP-based) and spectral (PLP-based) classifiers but also their complementarity. Then the combination of these two streams can bring an improvement in system performance. In the following experiments, the feature extraction framework is kept the same and only the TRAP processing (marked by dashed line in Fig. 8.3) is changed. Thus it is possible to evaluate influence of different TRAP processing on overall system performance.

The work of B. Chen has shown the importance of the temporal classifier. We will consider here two kinds of temporal classifiers: the standard 2-STAGE and HATS. The phoneme posteriors are estimated from block of 51 frames of log-critical band energies (`CRBE 51 frames`). Our baseline system (TRAP baseline) is a system with 2-STAGE temporal classifier which uses the `CRBE 51 frames` as input features. The size of *temporal classifier* is kept constant at 500 thousand weights (it is sum of all parameters of all NN used in the temporal classifier).

We first tested the Discrete Cosine Transform (DCT) TRAP processing. The 51 frames from one critical band are weighted by Hamming window and 1st to 25th DCT bases are applied. Thus the DCT also performs dimensionality reduction as in previous chapters (4, 7) leaving out the DC offset (0th DCT basis) – absolute energy level – and frequency components higher than 25 Hz. The resulting input to temporal classifier is denoted as `CRBE 51 DCT 25`.

Removing absolute energy level is beneficial when it can change in the test data i.e. due a different characteristic of speech transmission channel. When the data from the same source are used for both training and testing, which is our case, absolute energy level may contain useful information

classifier input	Temporal classifier	Male CVFA [%]	WER [%]
CRBE 51 frames	2-STAGE	64.6	34.6
CRBE 51 DCT 25	2-STAGE	65.0	34.7
CRBE 51 DCT 26	2-STAGE	65.1	34.5
CRBE 51 frames	HATS	65.6	34.2
CRBE 51 DCT 25	HATS	66.0	34.2
CRBE 51 DCT 26	HATS	66.1	**33.9**

Table 8.5: Results on CTS experiment for TRAP-based systems

classifier input	Temporal classifier	Male CVFA [%]	WER [%]
MCRBE 51 frames	2-STAGE	60.9	35.1
MCRBE 51 DCT 26	2-STAGE	60.2	35.4
MCRBE 51 frames	HATS	61.7	34.9
MCRBE 51 DCT 26	HATS	61.7	35.2

Table 8.6: Results on CTS experiment for MTRAP-based systems

for classification. The 0th basis was added to the DCT transform matrix capturing all components up to 25 Hz from input log-critical band energy in resulting vector (`CRBE 51 DCT 26`). Note, that mean and variance normalization over conversation side is done on log-critical band energies previously (Sec. 8.2.3).

The results for described inputs are shown in Tab. 8.5. The table shows cross-validation frame accuracy (CVFA) for male temporal classifier and the final word error rate (WER) of the LVCSR system.

We can see that both the frame accuracy of temporal classifier and the final WER are better when HATS temporal classifier is used instead of 2-STAGE classifier. Further, the improvement in cross-validation frame accuracy is observed when the log-critical band energies are transformed by the cosine transform. But the improvement does not carry on to the final WER for all cases. When 25 (1st to 25th) cosine bases are used, the final WER stays the same or degrades slightly. For the 26 DCT bases (0th to 25th) we see improvement for both temporal classifiers. The TRAP processing with 25 DCT bases is not tested further.

Considerable error rate reduction was achieved by combination of TRAP and MTRAP (modified TRAP) systems as described in Chapter 6. MTRAP-based features are obtained from processed – modified – critical band spectrogram (CRBS) as outlined in Sec. 5.1. The best modification for system combination turned out to be the frequency differentiation of the CRBS. 51 frames of the modified critical band energies (MCRBE) obtained by frequency differentiation are taken as input to temporal classifier (`MCRBE 51 frames`). We also processed the `MCRBE 51 frames` by the DCT transform where 26 bases were used (0th to 25th). These input features are denoted as `MCRBE 51 DCT 26`. Tab 8.6 shows system results when only MTRAP without and with DCT are used in the TRAP processing part of the system.

The natural choice of TRAP systems combination was a multi-stream combination since there already is the multi-stream combination in the Combined-Augmented feature extraction process. The posterior estimates obtained by MTRAP-based system create another stream to frame-wise multi-stream combination. Since there are three systems on the input of the multi-stream combination now, we denote the system as *3 way combination*. In our experiments, both TRAP- and MTRAP-based

classifier's input	Temporal classifier	WER [%]
CRBS 51 frames & MCRBS 51 frames	2-STAGE	34.7
CRBS 51 DCT 26 & MCRBS 51 DCT 26	2-STAGE	34.8
CRBS 51 frames & MCRBS 51 frames	HATS	34.0
CRBS 51 DCT 26 & MCRBS 51 DCT 26	HATS	34.4

Table 8.7: 3 way combination results on CTS experiment using the same processing and temporal estimator in both TRAP- and MTRAP- based systems

classifier's input	Temporal classifier	WER [%]
CRBS 51 frames & MCRBS 51 frames	2-STAGE	34.4
CRBS 51 DCT 26 & MCRBS 51 DCT 26	2-STAGE	34.4
CRBS 51 frames & MCRBS 51 frames	HATS	34.1
CRBS 51 DCT 26 & MCRBS 51 DCT 26	HATS	34.2

Table 8.8: Results on CTS experiment for systems with prior combination of temporal estimates

systems used the same TRAP processing of CRBE (i.e. using the DCT dimensionality reduction) and have the same temporal classifier. Tab. 8.7 shows results for the 3 way combination system. Note, that all three posterior estimators (PLP-, TRAP- and MTRAP- based) have 500 thousand weights so the total number of parameters in the system increased.

We observe that the combination of TRAP- and MTRAP- based features in one multi-stream combination block does not yield performance improvement over the temporal systems without critical band spectrogram modification (Tab. 8.5). Possible explanation is that the two temporal posterior estimates in the multi-stream combination overweight the PLP-based posteriors. Thus the resulting estimations tend to lean towards the temporal estimates and the benefit from the PLP ones diminishes. To verify our hypothesis, we performed the combination of outputs of TRAP- and MTRAP- based posterior estimations first. The temporal systems are combined using logarithmic average multi-stream combination (Sec. 6.1.2). Resulting temporal-based probability estimates are combined with PLP-based estimates using inverse entropy multi-stream combination as in previous cases. The results for the systems where the temporal estimates are combined prior the combination with PLP-based estimates are shown in Tab. 8.8.

Prior combination of the temporal systems brings performance improvement for most of the systems. However the performance of system with DCT processing of TRAP vector and HATS temporal classifier was not reached.

The vector concatenation was also a successful technique for combination of information from different sources. Here, the information from the original and modified CRBS is combined on the input of band specific estimator and the extraction of useful information for classification is left on the estimator. The resulting temporal estimates are then combined with PLP-based estimates by the multi-stream inverse entropy combination.

Processing of the critical band energies in our experiments is the same for both parts of the concatenated input vector (i.e. using dimensionality reduction or not). The 2-STAGE and HATS structure of temporal classifiers are tested and the classifiers have about 500 thousand weights. The results are shown in Tab. 8.9.

The vector concatenation technique is more efficient in extraction of relevant information for speech recognition from input features than the multi-stream combination. Improvement over the multi-stream combination with prior combination of temporal streams can be seen for most of the cases.

classifier's input	Temporal classifier	CVFA [%]	WER [%]
CRBS 51 frames & MCRBS 51 frames	2-STAGE	65.9	34.3
CRBS 51 DCT 26 & MCRBS 51 DCT 26	2-STAGE	66.1	34.4
CRBS 51 frames & MCRBS 51 frames	HATS	66.8	**33.8**
CRBS 51 DCT 26 & MCRBS 51 DCT 26	HATS	67.0	33.9

Table 8.9: Results on CTS experiment for systems with vector concatenation information combination

classifier input	Temporal classifier	Male CVFA [%]	WER [%]
CRBE 51 DCT 26	TMLP	66.56	34.4
CRBE 51 DCT 26	MLP4	68.76	**33.6**
CRBE 51 DCT 26 & MCRBS 51 DCT 26	MLP4	67.18	33.9

Table 8.10: Results on CTS experiment for one stage temporal estimators

Also, higher cross-validation frame accuracies over the systems without modified spectrogram input (Tab. 8.5) indicate higher portion of relevant information for phoneme classification. We achieved a slight improvement over `CRBE 51 DCT 26` with HATS temporal classifier in one case.

Based on the experiments, the spectrogram modification seems to be redundant in the *combined-augmented* feature extraction framework, where the role of different information source is played by the PLP-based estimates. The spectrogram modification tends to bring similar information as presented by the PLPs into the temporal system and the complementarity is partly lost. The resulting features are then inferior to the features where spectrogram modification is not used.

To test one stage estimators, tonotopic multi-layered perceptron (TMLP) and 4 layer perceptron (MLP4), the (best performing) `CRBE 51 DCT 26` input is used. The final WER of the system and the cross-validation frame accuracy of the male temporal classifier are shown in upper part of Tab. 8.10. Although the TMLP classifier was reported to be the best for `CRBE 51 frames` in [15], better performance for temporal processing with dimensionality reduction was obtained by the MLP4 estimator. The TMLP estimator is slightly inferior to the HATS structure in this case. This opposite behavior of classifiers performance may be caused by the preprocessing of temporal trajectories by DCT which encodes the information within a critical band. Then the need of individual band processing may diminish.

Finally, we tested also the vector concatenation technique with the MLP4 classifier[4]. The results for the system are shown in the lower part of Tab. 8.10.

8.3 Conclusions and discussion

In this chapter, we introduced different TRAP techniques to LVCSR. For meetings, where we developed the whole recognition system, we first obtained the best TRAP-based features and then combined them with standard spectral-based features. For continuous telephone speech (CTS) recognition, we introduced different TRAP processing into an already existing system. In both cases, augmenting TRAP-based features to spectral-based features brings considerable reduction of word error rate over the spectral-based features alone.

[4]We could not test the TMLP classifier due the limitation of the neural net training software.

While testing different TRAP-based features and their combination with standard cepstral features, two important points appear:

1. **Proper combination technique of probabilistic and cepstral features.** The experiments on meetings show that using simple technique may not lead to system improvement. When more complex discriminative technique for dimensionality reduction is used instead of simple principal component analysis, significant improvement is achieved.
   For CTS experiment, the combination of posterior features with cepstral based features was largely optimized by ICSI researches B. Chen and Q. Zhu. There, the dimensionality of posterior based features is first sufficiently reduced and resulting vector is then concatenated with spectral based features. Thank to this optimization, we did not encounter the problems of improper feature combination as for meeting recognition.

2. **Complementarity of the features.** This point was not our concern in the meetings experiment where the TRAP-based features were the only posterior features in the system. But it fully arises in CTS experiments where we combine TRAP-based posterior estimates with those based on PLP features, and the resulting posterior based features are combined with cepstral features. Here we observe that the spectrogram modification which was largely beneficial in previous experiment is almost redundant, because the information about the spectrogram shape along the frequency axis is presented by the PLP features and posteriors derived from them.

We observed consistently better cross-validation frame accuracy and system performance when HATS temporal estimator was used instead of 2-STAGE classifier. Further, the improvement due to the processing of critical band energies (decomposition of the trajectory to DCT bases) carries through the different temporal classifiers.

The interesting observation is that the WER of systems with `MCRBS 51 frames` are consistently slightly lower than the final WER of systems with `MCRBS 51 DCT 26`. This suggests that the dimensionality reduction of TRAP modified by frequency differentiation may not be optimal. This gives us space for further improvement when different processing would be used for different streams.

Chapter 9

Conclusion

The chapter will first summarize current research of other researchers related to TRAP technique. The researchers usually targets only one part of the TRAP-based probabilistic feature extraction. Published techniques suggested for derivation of TRAP vectors are described in Sec. 9.1.1, proposed parameters (features) derived from TRAP vectors are described in Sec. 9.1.2, classifiers used to estimate phoneme posteriors from TRAP features are discussed in Sec. 9.1.3. Then, techniques originated from TRAP system but making more essential changing to it such are UTRAP (Sec. 9.1.4), split-context (Sec. 9.1.5) and spectro-temporal patterns (Sec. 9.1.6) are described.

The work presented in this thesis is summarized in Sec. 9.2, highlighting the benefits for current state-of-the-art recognition systems and also the ideas adopted by other researchers. Finally, the work is concluded and possible directions of future research in TRAP-based probabilistic features are drawn.

9.1 Current work related to TRAP

Here, the overview of current work of other researchers related to TRAP-based feature extraction will be given with short description of the ideas, experimental setups and results.

9.1.1 Derivation of TRAP vectors

The first question we may have in mind when working with TRAP-based system is, whether the TRAP vectors are obtained in an appropriate way. We work with time evolution of energy in one critical band. However, this energy is obtained through spectrum computed over 25ms of speech signal. As an alternative, several approaches for direct extraction of critical band energy trajectory directly from the waveform signal were suggested:

9.1.1.1 Time-domain derivation of TRAP

Motlíček [68] is proposing the derivation of temporal patterns directly in time domain. The speech signal is first filtered by a bank of gamma-tone filters. The spectral properties of the filter set are similar to those of critical band filters. As a result, a set of filtered time-domain signals are obtained, each of which has its spectrum in a band given by one band-pass gamma-tone filter. The second step in this derivation of TRAP vectors is to move spectrum of each signal towards zero. This is done by multiplication of the filtered time signal with complex exponential with frequency corresponding to the center of the gamma-tone filter (a demodulation). Resulting signals are then filtered by a low-pass filter to obtain the modulation signal. This modulation signal has only low-frequency components and can by largely down-sampled without loosing important information. In the presented experiment, the new sampling frequency was 80Hz. Finally, the temporal patterns of length 500 ms (40 samples)

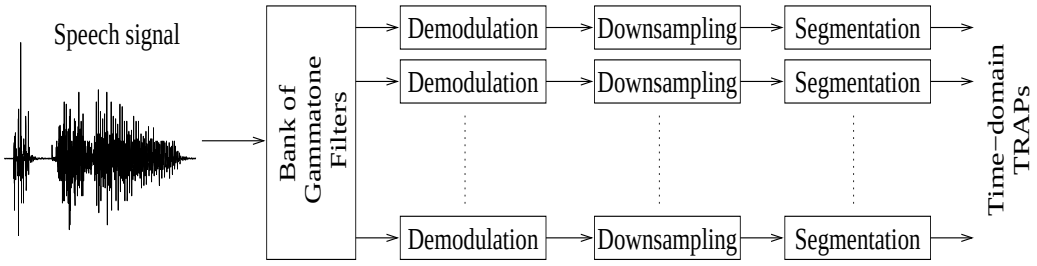


Figure 9.1: Block diagram of time-domain TRAP vectors extraction

are extracted each 10 ms following the segmentation of standard features. The block diagram of the time-domain TRAP extraction is shown in Fig. 9.1.

Reported results are frame classification accuracies on frequency-conditioned and merger classifiers. The frequency-conditioned classifiers are trained on TIMIT database, merger is trained on OGI-Stories database. The length of time-domain TRAP vector is 40 points, length of standard TRAP is 51 points. At given level, both, "classical" and time-domain, TRAP vector extraction techniques perform about the same.

9.1.1.2 LP-TRAP

Athineos and Ellis [3] proposed another way to describe temporal evolution of signal energy in sub-bands without initial short-time framing. Their method is based on the duality of linear prediction (LP). In the same way we model the envelope of power spectrum using LP in time, the square of temporal Hilbert envelope can by modeled by using LP in frequency. When LP is applied in frequency domain, the poles of the model describe temporal peaks. When window of several hundred milliseconds is used, the LP procedure automatically decides how to distribute the poles to best model the temporal structure within the window. The frequency-domain representation of the speech signal is provided by the discrete cosine transform (DCT).

In the following paper [4] authors together with Hermansky extend the work to the TRAP domain by introducing frequency sub-bands. To get an all-pole approximation of the Hilbert envelope in a specific sub-band, the prediction needs to be derived only from the appropriate part of signal's frequency representation. The frequency parts were obtained from the DCT representation of the signal by applying 15 overlapping Gaussian-shape windows spaced linearly on Bark frequency axis similarly to critical band processing. Then the LP was applied on each band. Finally the cepstral recursion is used to convert the all-pole model of the temporal trajectories to modulation spectra. To obtain results comparable to standard TRAP approach, the length of the signal was set to one second and the process was repeated with 10 ms shift to obtain the same feature rate as in usual processing.

The results are reported on continuous digit recognition task with OGI-Numbers database similar to the one described in Chapter 3, where the band-conditioned classifiers are trained on OGI-stories database and the merger and HMM models are trained on OGI-numbers. Direct comparison of results is not possible due to the difference in the HMM recognizer. Authors report improvement over the standard TRAP system.

9.1.1.3 Fepstrum

Fepstrum representation of speech signal is proposed in [86] by Tiagi and Wellekens. The approach is similar to the one proposed by Motlíček described in section 9.1.1.1. The authors view speech as a series of amplitude modulated (AM) signals $s_{AM}(t) = s_M(t)c_0(t)$ where s_M is a low frequency signal used for modulation and c_0 is a carrier signal. Since the s_M has only low frequency components, it is necessary to use a narrow band-pass filter to separate one AM signal from the others. Then, the

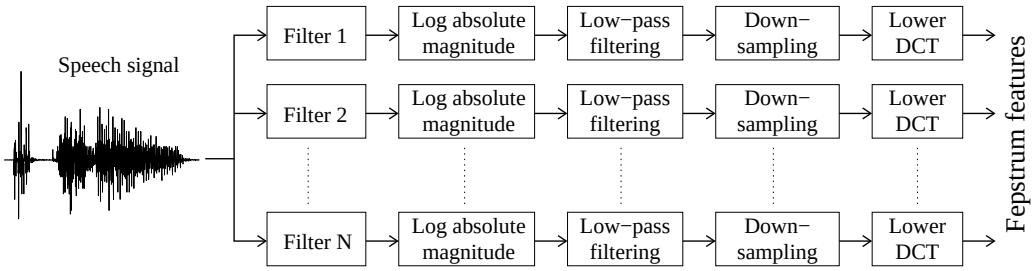


Figure 9.2: Block diagram of fepstrum TRAP vectors extraction

modulation signal s_M is estimated from the obtained narrow band signal s_AM. The s_M can be seen as a critical band energy trajectory in the TRAP approach. The modulation signal is then down-sampled and transformed by DCT and only lower DCT coefficients are retained as feature vector. This representation of input signal is called *fepstrum* to distinguish from previously used *modulation spectrum*. The fepstrum features from all bands are concatenated and KLT is used to decorrelate them and to reduce their dimensionality. Resulting vectors are then fed into MLP, which is trained to classify phonemes, to obtain fepstrum-based probabilistic features. The block diagram of fepstrum-based feature extraction is shown in Fig 9.2.

The experiments were carried on numbers recognition task (31 words). The fepstrum features were extracted using 20 linearly spaced band-pass filters with the bandwidth of 200Hz. The AM signal is obtained as the logarithm of absolute magnitude of the filtered signal. The AM signals are then filtered with low-pass filter with cutoff frequency 200Hz and are down-sampled by factor of 40. The framing is done using rectangular window of 85 ms with time shift 10 ms. Each frame is then represented by first five DCT coefficients resulting in a vector with 100 (20×5) coefficients. This vector is decorrelated by KLT and its dimensionality is reduced to 60 coefficients. These features are fed to MLP classifier to obtain probability estimates of phonemes which are again transformed by KLT to get 27 fepstrum-based probabilistic features. Authors report improvement when these features are concatenated with standard MFCCs.

9.1.2 Parametric representation of TRAP

The appropriate representation of temporal trajectories of critical band energies is another point of interest in TRAP system. The goal of parametric representation of TRAP vectors is to yield higher classification accuracy or to get better properties than the original representation from classification point of view: lower dimensionality of final vector and/or its capability to generalize over different data. Today, the mostly used representation are the DCT coefficients. However, the way to this representation was not so straightforward. Motlíček [68] proposed several representations of temporal trajectories.

He firstly attempted to represent the spectrum of TRAP vector (i.e. modulation spectrum in given band) by Linear Prediction Coefficients (LPC). This approach failed completely, therefore a step back was done - investigation of TRAP vector representation itself instead of representing its spectrum. He tested several forms of Discrete Fourier Transform (DFT) coefficients – absolute values only, absolute values and phase and real and imaginary components. These experiments show the importance of phase information which is necessary for reconstruction of original temporal pattern and which was lost in LPC coefficients. Further, two different definitions of Discrete Cosine Transform (DCT) were tested showing basically no difference between them. Finally, Principal component analysis and Discrete Hadamard Transform (DHT) were proposed for TRAP representation.

Reported results are frame classification accuracies on frequency-conditioned and merger classifiers. The frequency-conditioned classifiers are trained on TIMIT database, the merger is trained on OGI-Stories database. The length of TRAP vector is 40 points. The size of each parametric representation

is kept the same as size of TRAP vector, i.e. 40 coefficients.

It is hard to compare proposed representations at the frame classification accuracy level for several reasons: First, the reported results span small range so the changes are rather slight. Then, the neural net initialization plays certain role in final frame classification accuracy. Last, the changes on classification accuracy do not always carry through to further processing (speech recognition).

It is possible to group the proposed parametric representations into three categories:

- With loss of information, such as LPC or absolute values of DFT, where the important information (phase in this case) for phoneme classification is lost, resulting in very poor classification accuracies.

- Without loss of information such as full DFT (real and imaginary part), DCT, PCA and DHT. All these representations perform about the same.

- With possible dimensionality reduction. These are PCA, DCT and full DFT (real and imaginary part).

Recently, Hermansky and Fousek [35] proposed another representation of TRAP vector called Multi-resolution RASTA (MRASTA). The transform of TRAP vector into a set of new parameters is seen as a filtering of critical band energy trajectory by a set of filters. The filter responses are given by first and second order derivatives of Gaussian function with different variances which cause different frequency resolutions of the filters.

Authors report results on continuous digit recognition task with OGI-Numbers database similar to the one described in Chapter 3. Direct comparison of results is not possible due to the difference in the HMM recognizer. Improvement over the baseline TRAP system is reported, but comparison with any of the previously proposed TRAP representations (such as DCT or PCA) is not given. According to our internal tests made on CTS task (Section 8.2), the difference of performance of DCT and Gaussian TRAP features is insignificant (TRAP-DCT features are slightly better).

9.1.3 Temporal classifiers

B. Chen in his work [13] developed better TRAP-based estimator than the standard 2-STAGE posterior estimator. The summary of the work is given in 8.2.4. Shortly, two main questions led the research:

1. Can we skip the mapping from the output of the matched filters to critical band phone posteriors?

2. Is there a better way to learn critical band matched filters? (where the term "matched filters" is used for the weights from input to hidden layer)

Pursuing the first question led to removing parts of the band-conditioned neural net. The best results were obtained when outputs of hidden layer formed the merger input vector [14]. This structure of temporal classifier is called Hidden Activation TRAPS (HATS) and is shown in Fig. 8.5.

Second question led to integrating the band-conditioned structure in one neural network. The resulting 4-layer MLP is called Tonotopic Multi-Layer Perceptron (TMLP) [15]. The first hidden layer of TMLP consists of several groups of hidden units, each one is constrained to receive the input only from a single critical band. This represents the band-conditioned step. The second hidden layer is fully connected and represents the merger. The TMLP structure is shown in Fig. 8.6.

The results are obtained on LVCSR task with conversational telephone speech. The TRAP-based posteriors are estimated from plain critical band energies and they are combined with the PLP-based posteriors. Resulting posterior features are concatenated with spectral PLP-based features.

WER improvement over standard 2-STAGE temporal classifier was seen in both cases. Better performance was obtained with the TMLP temporal classifier.

9.1.4 UTRAP

The Universal TRAP (UTRAP) approach was proposed by Hermansky and Jain in [36]. The idea was inspired by the observation of mean TRAP vectors for all phonemes over all critical bands. It was possible to find similar patterns for different phonemes in different frequency bands. So if we had some set of basic patterns, it would be possible to train a single – universal – classifier which would be used in all bands. The combination of basic patterns found in different bands would be converted to phoneme classes by merger. This approach significantly simplifies the temporal classifier in TRAP system. In addition to having only one band-conditioned classifier, it brings also reduction of the size of merger in case the number of universal patterns is smaller than the number of phonemes.

The mean TRAPs of all phonemes from all classes were clustered into nine universal classes. Authors used agglomerative hierarchical clustering technique with correlation measure to obtain new classes. The outcome of this clustering was a table where each phoneme in each critical band was associated with one universal class. According to this table, the training data were relabeled and a universal band-conditioned classifier was trained. Note that the same phoneme in different critical bands could be associated with different universal class.

The experiments with this system were carried out on the Stories – Numbers experiment (same setup as in Chapter 3) and on Aurora 2 and 3 databases. The UTRAP system was compared with TRAP system where broad phonetic classes were used as targets for band-conditioned classifier. The performance of both systems is about the same, but both are worse than TRAP system with phonemes as classes in both stages. When the probabilistic features were concatenated with MFCC, UTRAP based features consistently gained better performance over TRAP with broad phonetic classes as targets in the first stage.

The technique was revised by Svojanovský in [85] on the Stories – Numbers experimental setup. Nine universal classes were used also here although it was shown that error can be lower for higher number of classes. The author proposed to train the universal band-conditioned classifier only on part of the data coming from 3^{rd} to 5^{th} critical band, where most speech energy is present. This innovation brought quite nice improvement over the system, where universal band-conditioned classifier was trained on data from all bands. Further, three clustering techniques were tested with almost the same results which means that the actual clustering doesn't play important role. Finally, the author suggested one band-conditioned classifier trained on one band data with phoneme classes to be used in all bands. This approach brought improvement over the standard TRAP system.

9.1.5 Split context

P. Schwarz in [76] proposed splitting the block of critical band energies to temporally separated parts. The motivation is that typical longer patterns are occurring less frequently than shorter ones. This can be illustrated on percentage of unseen N-grams from test data in training data. If we choose phoneme 1-grams, the test set is fully covered. If we use 3-grams, then some percentage of 3-grams in the test set is not seen in the training set (for TIMIT database it is 18.8% according to [76]). If we would like to decrease the number of unseen N-grams, we should step back to lower order N-gram. This is possible by splitting the context to several parts and then combining the results from each part.

After splitting the pattern into left and right contexts, only one side context is considered in one classifier. As the context is shorter, also the number of unseen "N-grams" is smaller and the "time-conditioned" classifier can better classify the unseen test vectors. The merger then combines the time-conditioned estimates. As arising from the above, split context temporal classifier is a 2-STAGE classifier where classifiers in the first stage are conditioned in time instead of frequency. The scheme of the split context system is shown in Fig. 9.3.

The second modification was the change of classifier targets. HMM phoneme models usually consist of three states – left, center and right – each having capability to model part of the phoneme. To

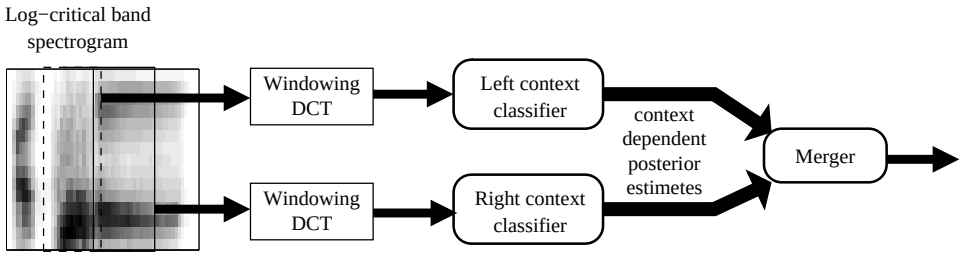


Figure 9.3: Block diagram of split context 2-STAGE phoneme posterior estimator

imitate this state property of HMM models, the classifier outputs are associated with phoneme states. Thus the input patterns, which may be different for the beginning and the end of the phoneme, can be better classified.

Schwarz evaluated the proposed modifications on phoneme recognition of TIMIT database. The input for the classifier are 31 frames of critical band energies, where each band is weighted by a triangular window and transformed by DCT. The only criterion here is the phoneme error rate (PER), so the sizes of classifiers are tuned to get the best PER. The improvement for split context system and phoneme-state outputs is shown. The following work [77] introduced further possibility of splitting the spectro-temporal plane of critical band energies. The best PER was obtained by splitting the block into 5 time-conditioned parts.

9.1.6 Spectro-temporal patterns

The idea of spectro-temporal patterns originates from efforts to optimize the steps of feature extraction on data. Thus we can see data-driven RASTA processing [88] where time sequences of features were analyzed using LDA and the first eigen vector taken as FIR filter impulse response. The attempts to replace the DCT decorrelation of critical band energies by bases obtained from PCA or LDA (with phoneme classes) of spectral vectors were described in [38, 32]. Also, Independent Component Analysis (ICA) has been used on top of critical band energies or MFCCs [72].

When LDA is applied on vector formed by several concatenated feature vectors, the resulting bases can be viewed as two-dimensional bases. The two-dimensional LDA bases computed from 101 frames of 15 critical bands energies and phoneme classes were studied in [49]. Authors showed the possibility of constructing two-dimensional bases by combination of bases derived independently in spectral and temporal domains. Recently, Valente and Hermansky returned to the idea of linear discriminant analysis of spectro-temporal plane [87]. They analyzed 30 hours of conversational telephone speech with 40 phoneme classes. The analyzed input was a block of 101 frames of 129 linear predictive (LP) power spectrum coefficients.

An independent analysis of spectral and temporal domain was done as well as the analysis of two-dimensional patterns. The outputs of one-dimensional analysis supported the earlier findings, and also the outcome of two-dimensional analysis was the same as in [49] – a two dimensional basis can be composed from spectral and temporal one. When these bases were used for feature extraction (projecting the LP spectrum on 13 spectral bases and then on 3 temporal) the improvement over PLPs was only marginal. The results from direct use of two-dimensional bases were not reported.

The two-dimensional basis can be seen as a time-frequency filter or a spectro-temporal pattern we are looking for in an input block of critical band energies. Kleinschmidt and Gelbart [51] used Gabor filters (two-dimensional sinusoid weighted by Gaussian envelope) as spectro-temporal patterns. The need for diagonal spectro-temporal patterns for speech recognition, which are not present in LDA bases, is supported not only by observation of speech signal (formant's transitions) but also by examination of auditory cortex of different mammal species, where large percentage of neurons respond differently to upward versus downward-moving ripples in the spectrogram [18]. Since there is very high

number of possible Gabor filters which can be used, the number of free parameters was restricted. The "optimal" set of Gabor filters was selected automatically – a simple linear classifier was used to evaluate the importance of each individual Gabor filter based on its contribution to classification performance.

Three sets of 60 filters were designed and optimized for broad phonetic categories, phonemes and German digit words. The authors reported the portion of filters with purely spectral, purely temporal and spectro-temporal modulation which is 35/23/1 for broad phonetic categories, 34/22/4 for phoneme classes and 12/18/30 for word classes. This supports the conclusion from LDA bases study where independence of spectral and temporal domain was observed for phoneme classes. In accordance with the articulatory cortex study, it suggests, that some neurons are trained to act for more complex sounds (for humans perhaps names and common words).

The derived phoneme Gabor filters were applied on 23 Mel scaled critical band energies and the resulting features were on-line normalized. First and second derivatives were computed resulting in 180 features. These were fed into MLP trained to classify 56 phonemes. MLP outputs were decorrelated using KLT and were given to GMM-HMM recognizer. The improvement over baseline MFCC was seen for Gabor-based probabilistic features. Improvement was also seen when these features were combined with Qualcomm-ICSI-OGI features [1] on noisy digit recognition from AURORA 2 database.

9.2 Summary and conclusions

The work presented in this Ph.D. thesis is direct extension of previous works of S. Sharma [78] and P. Jain [45]. The work concentrates on the input features to the temporal classifier derived from critical band trajectories.

The first part of the work described various experiments performed on Stories-Numbers continuous digit recognition experimental setup (Chapter 3). The dimensionality reduction of TRAP vector using DCT, PCA and LDA was tested first and compared with the TRAP baseline. The dimensionality reduction using DCT and PCA brought improvement in system performance. The LDA dimensionality reduction did not achieve the baseline performance which we think is because of poor estimation of LDA bases.

Then we studied the *band merging* system where TRAP vectors from several critical bands created input to one frequency-conditioned probability estimator. This *band merging* system was also tested with dimensionality reduction performed separately on TRAP vector from each band before their concatenation, or jointly on the resulting concatenated vector. All these modifications brought improvement in system performance. The dimensionality reduction further improved performance of the system with concatenated bands. Dimensionality reduction on concatenated TRAP vectors achieved better performance than concatenation of independently processed vectors. We analyzed the dimensionality reduction bases to find out, why dimensionality reduction applied on concatenated vectors performed better. We found, that PCA bases computed from three concatenated TRAP vectors contain bases which performs differentiation and double differentiation over the frequency and thus encode also the frequency relations between neighbouring critical bands.

Next (Chapter 5), simple modifications of critical band spectrogram prior to deriving the TRAP vector were proposed. One-dimensional operations performing averaging and differentiating of three adjacent points in either time or frequency domain and two-dimensional Sobel filters of size 3×3 were tested. These were motivated by the nature of observed PCA bases. But it turned out, that a system with only one kind of spectrogram modification cannot outperform the complex behavior of whole PCA transform as one operator can cover only one aspect of this transform.

That was the reason why next Chapter 6 was devoted to the study of TRAP systems combinations. The TRAP systems can by combined at three different places:

- On the end of the processing. The final probabilities can be combined by multi-stream combi-

nation techniques. We evaluated three of them:

  - Averaging combination, where class probabilities from different systems were averaged.
  - Log-averaging combination, where class probabilities from different systems were averaged in log domain.
  - Inverse entropy combination with static threshold, where weighted average of classes probabilities was computed. The weight for individual stream was assigned dynamically according to the entropy of given output.

  This technique requires to train full "component" systems. It can be beneficial when the systems are already trained or when there is a need to have also outputs from each individual system.

- In the middle of TRAP processing. The band-conditioned estimates from different streams are combined. But since the accuracy of band-conditioned classifiers is rather low, the result of this combination are not final probability estimates. This kind of combination is a preprocessing step before the partial probabilities are fed into a merger classifier. The following techniques were tested:

  - Averaging combination. We averaged the outputs belonging to the same critical band but from different systems, or the outputs from the same system but from different bands. The number of output partial estimates entering the merger was kept at 15 in both cases, just as for any other merger.
  - Weighted averaging combination. This combination is the same as above but the individual systems were weighted according to their classification accuracy.
  - PCA based combination. For this kind of combination all outputs for given class formed a vector – *class vector*. The PCA was obtained from this vectors, so each class had its own PCA transform. Only 15 coefficients were returned for each class to obtain the same size of merger input vector.

  Combination of band-conditioned outputs requires to train the band-conditioned estimators for combined systems and a common merger.

- At the beginning of the posterior estimation. We can directly concatenate different TRAP-based features on the input to band-conditioned classifier. This technique requires one TRAP system to be trained on concatenated input vectors.

The combinations of TRAP systems with differently modified critical band spectrogram was evaluated. Very good results were obtained whenever a system with critical band spectrogram modified by some form of frequency differentiation appeared in the combination. When the baseline system is combined with system with frequency differentiation, all combination techniques outperform the best results seen so far – concatenation of TRAP vectors from three bands followed by dimensionality reduction. The best performing combination technique was log-averaging multi-stream combination followed by vector concatenation.

The following part of the work (Chapter 7) examined the behavior of TRAP systems in noisy conditions. The experimental setup was the reference AURORA 2 recognition setup. We used clean training data for training our systems. Not all systems presented so far were evaluated on noisy data, but only those which improved the performance of the baseline system.

We studied the effect of different normalisation schemes on the performance of TRAP-based system first. We evaluated sentence and TRAP based mean and mean+variance normalization against the case where no normalization was used. The mean normalization failed. This is because the additive noise, presented in test sets, increased the energy level in critical band spectrogram. Since the energies are presented in log domain, the noise affects more silent parts of the utterance, which leads to change

of the dynamics in TRAP vector. Since the system was trained on clean data, it fails to recognize patterns with their dynamics flattened by noise. The mean+variance normalization brought noise robustness to the system. The variance normalization kept the dynamic range or the critical band energy the same over different noise levels so the classifier was able to make the correct decision.

We proceeded further with the noise robust normalisation scheme and tested dimensionality reduction of TRAP vectors in noisy condition. Dimensionality reduction showed improvement for test sets with SNR$\geq$15dB. In test sets with stronger noises, an impairment was observed. We explain this by the ability of neural network to generalize better over actual energy trajectory than over its representation. Such representation is also more vulnerable to noise - if somewhere in the energy trajectory a sudden peak caused by noise appears, this isolated event will be spread over all vector coefficients after dimensionality reduction.

Further, merging of three consecutive bands into one frequency-conditioned classifier was tested. This techniques also improves system performance for SNR$\geq$15dB but hurts for lower SNR. This is caused by the presence of more complex pattern on the classifier input, which is easier to be corrupted by noise. When dimensionality reduction was applied on the concatenated TRAP vectors, the deterioration for low SNRs increased as the disadvantages of both techniques for noisy data summed up.

The systems with critical band spectrogram modified by frequency differentiation were trained and their combination with system based on original critical band spectrogram were evaluated. The modified TRAP system has rather poor performance but the multi-stream combination with original system brought considerable improvement for test sets with SNR$\geq$10dB. When we examined the function of the static threshold in inverse entropy combination technique, we were able to find a global optimum for all SNRs, which brought further improvement. The system with vector concatenation technique further improved results for SNR$\geq$0dB and performed the best on this experimental setup.

Last, in Chapter 8, TRAP-based probabilistic features were used in two LVCSR tasks. We tested the performance of TRAP-based probabilistic features alone and in combination with spectral-based features MFCC and PLP. In the first experiment with meeting speech recognition, we first focus on finding the optimal TRAP vector processing. As the evaluation of different processing on GMM-HMM recognition system would take enormous amount of time and computation resources, a monophone hybrid recognition system was build for this optimisation step. A system with vector concatenation combination of TRAP vectors from original and frequency modified critical band spectrograms was found optimal. The optimal length of TRAP vector was found to be 41 frames and normalisation scheme providing the best performance was utterance-based mean normalization. These TRAP vectors were transformed by DCT. These optimal TRAP-based probabilistic features were then decoded by full GMM-HMM recognition system. We observed the same performance for TRAP-based features and MFCC features when word-internal context dependent phoneme models were used. The TRAP-based features stayed behind MFCC when more complex word-external models were used and also when speaker adaptation technique was used.

The combination of TRAP-based and MFCC features was tested further. We concatenated both features (after their global mean and variance normalization) and performed PCA on top of it. Different size of resulting vectors were tested. Best results were obtained with the resulting size of 40 coefficients. But this combined features did not achieve the performance of PLP features. That was why we decided to use more sophisticated HLDA transform which is looking for discriminative information in the data. With HLDA transformation, the resulting feature size was 39 coefficients and the results improved over PLP in all cases.

The effect of different processing of TRAP vector on system performance was further studied on Conversational Telephone Speech recognition task. The TRAP-based posterior features were merged not only with spectral-based PLP features but also with probabilistic features based on nine frames of PLP features in this experimental setup. Thus the complementarity of TRAP based features was tested not only to cepstral features but also to a different probabilistic features. Also, different

temporal classifiers were examined under this setup. The DCT dimensionality reduction improved the baseline performance. Adding frequency differentiation spectrogram modification and its combination with original TRAP system did not show such great potential as in previous cases. This is due to the PLP-based probabilistic features which already provide similar information as TRAP with spectrogram modification. As for different temporal classifiers, the HATS neural net structure and four layer neural net were giving consistently better results then original TRAP neural net structure. The best results were obtained with TRAP features with DCT dimensionality reduction and four layer neural net as temporal classifier.

9.3 Future plans

The future work is planned to focus on combination of our experience with the current work of Petr Schwarz, mainly his innovation of phoneme states as neural nets targets. The choice of neural net outputs was not questioned in our work and further benefits can be gained by incorporating phoneme states or other target classes into our system. Further, we will focus on the comparison of tested temporal classifiers with the split context one.

Thanks to the development of new software SNet [53] incorporated into Speech@FIT's STK toolkit, handling of neural networks is now much easier. The main benefit of this tool to our work is that it limits the necessity of creating large input and output files for neural nets which was a big disadvantage of the QuickNet software. It implements neural net as a general transformation which can be easily used. The following possibilities are open:

- Creation of neural nets with arbitrary numbers of hidden layers and complex structures building on and extending the investigated HATS or TMLP approaches.
- Joint training of neural nets and hidden Markov models under Maximum Likelihood (ML) or discriminative criteria.

We performed preliminary experiments [31] and plan to continue the work in feature extraction directly by neural network while skipping the conversion of class posteriors to features. It is possible to use a neural net with 5 layers with a bottle-neck in the middle layer. Such bottle neck generates features suitable for subsequent processing by GMM-HMM recognizer.

We will be also aware of progress of other laboratories in TRAP vector derivation techniques. In case a technique brings improvement in terms of final word error rate or processing time, we will test it and eventually include it in our system. On the other hand, we are cooperating with other institutions using TRAPs and posterior features such as IDIAP, ICSI, SRI or LIMSI, and we believe our research and development can be useful also for their work.

9.3.1 The final point

During our work, we have assisted at the development of TRAPs, beginning as a novel technique no one really believed in, into a paradigm used in state-of-the-art experimental systems for meeting recognition [83, 23, 47] that are finding their way into commercial systems[1]. I am personally happy that I could participate on this challenging research.

[1]For example the phoneme recognizer distributed by Speech@FIT's spin-off Phonexia: http://www.phonexia.com.

Bibliography

[1] A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas. Qualcomm-ICSI-OGI features for ASR. In *Proc. ICSLP 2002*, Denver, Colorado, USA, 2002.

[2] A.G. Adami, R. Mihaescu, D.A. Reynolds, and J.J. Godfrey. Modeling prosodic dynamics for speaker recognition. In *Proceedings ICASSP '03*, pages IV – 788 – 791, Hong Kong, China, 2003.

[3] M. Athineos and D.P.W. Ellis. Frequency-domain linear prediction for temporal features. In *Proc. of IEEE ASRU*, pages 261–266, St. Thomas, U.S. Virgin Islands, Dec 2003.

[4] M. Athineos, H. Hermansky, and D.P.W. Ellis. LP-TRAP: Linear predictive temporal patterns. In *Proc. ICSLP 2004*, pages 949–952, Jeju Island, KR, October 2004.

[5] L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.*, 37:1554–1563, 1966.

[6] C. Benitz, L. Burget, B. Chen, S. Dupond, H. Garudadri, H. Hermansky, P. Jain, S. Kajarekar, and S. Sivadas. Robust ASR front-end using spectral-based and discriminant features: experiments on the AURORA task. In *Proc. Eurospeech 2001*, Aalborg, Denmark, September 2001.

[7] M. Berouti, R. Schwartz, and J. Makhoul. Enhancement of speech corrupted by acoustic noise. *IEEE Trans. Acoust., Speech, Signal Processing*, 27:208–211, April 1979.

[8] J. A. Bilmes. Maximum mutual information based reduction strategies for cross-correlation based joint distribution modeling. In *Proc. IEEE ICASSP 98*, pages 469–472, Seattle, WA, May 1998.

[9] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust., Speech, Signal Processing*, 27:113–120, April 1979.

[10] H. Bourlard and N. Morgan. *Connectionist Speech Recognition: A Hybrid Approach.* Number 247 in Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, 1994.

[11] L. Burget and H. Hermansky. Data driven design of filter bank for speech recognition. In *Text, Speech and Dialogue : 4th International Conference, TSD 2001*, pages 299–304, Zelezna Ruda, Czech Republic, Sep 2001.

[12] B. Chan, S. Chang, and S. Sivadas. Learning discriminative temporal patterns in speech: Development of novel TRAPS-like classifiers. In *Proc. Eurospeech 2003*, Geneva, Switzerland, 2003.

[13] B. Chen. *Learning Discriminant Narrow-Band Temporal Patterns in Automatic Recognition of Conversational Telephone Speech.* PhD thesis, University of California, Berkeley, Berkeley, CA, USA, 2005.

[14] B. Chen, Q. Zhu, and N. Morgan. Learning long-term temporal features in LVCSR using neural networks. In *Proc. ICSLP 2004*, Jeju Island, KR, October 2004.

[15] B. Chen, Q. Zhu, and N. Morgan. Tonotopic multi-layered perceptron: A neural network for learning long-term temporal features for speech recognition. In *Proc. ICASSP 2005*, Philadelphia, PA, USA, March 2005.

[16] R. Cole, M. Noel, T. Lander, and T. Durham. New telephone speech corpora at CSLU. In *Proc. of EUROSPEECH 1995*, pages 821–824, Madrid, Spain, 1995.

[17] M. Cooke, A. Morris, and P. Green. Missing data techniques for robust speech recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 863–866, Munich, April 1997.

[18] D.A.Depireux, J.Z. Simon, D.J. Klein, and S.A. Shamma. Spectro-temporal response field characterization with dynamic ripples in ferret primary auditory cortex. *J. Neurophysiol.*, 85:1220–1234, 2001.

[19] D. Ellis and N. Morgan. Size matters: An empirical study of neural network training for large vocabulary continuous speech recognition. In *Proc. ICASSP 1999*, pages 1013–1016, Phoenix, Arizona, USA, March 1999.

[20] D. P. W. Ellis and M. J. Reyes Gomez. Investigations into tandem acoustic modeling for the Aurora task. In *Proc. Eurospeech 2001*, Aalborg, Denmark, sep 2001.

[21] D. W. P. Ellis, R. Singh, and S. Sivadas. Tandem acoustic modeling in large-vocabulary recognition. In *Proceedings of ICASSP'01*, Salt Lake City, Utah, USA, May 2001.

[22] D. P. W. Ellis et al. SPRACH: Speech recognition algorithms for connectionist hybrids, SPRACH core package. http://www.icsi.berkeley.edu/~dpwe/projects/sprach/sprachcore.html.

[23] T. Hain et al. The AMI system for the transcription of speech meetings. In *Proc. ICASSP 2007*, pages 357–360, Hononulu, US, Apr 2007. IEEE Signal Processing Society.

[24] M. Finke and A. Waibel. Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition. In *Proc. Eurospeech '97*, pages 2379–2382, Rhodes, Greece, 1997.

[25] H. Fletcher. *Speech and Hearing in Communication*. D. Van Nostrand Company, Inc., New Jersey, Nov 1953.

[26] G.D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61:268–278, Mar 1973.

[27] K. Fukunaga. *Introduction to Statistical Pattern Recognition,(2nd ed.)*. Academic Press Professional, Inc. San Diego, CA, USA, 1990.

[28] J.S. Garofolo, L.F.Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, and N.L. Dahlgren. DARPA–TIMIT acoustic–phonetic speech corpus. Technical Report NISTIR 4930, U.S. Department of Commerce, National Institute of Standards and Technology, Computer Systems Laboratory, February 1993.

[29] L. Gillick and S.J. Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89.*, pages 532–535, Glasgow, UK, May 1989.

[30] S. Greenberg, T. Arai, and R. Silipo. Speech intelligibility derived from exceedingly sparse information. In *5th International Conference on Spoken Language Processing (ICSLP)*, pages 2803–2806, Sydney, Australia, Dec 1998.

[31] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký. Probabilistic and bottle-neck features for LVCSR of meetings. In *Proc. ICASSP 2007*, pages 757–760, Hononulu, US, Apr 2007. IEEE Signal Processing Society.

[32] R. Heab-Umbach and H. Ney. Linear discrimination analysis for improved large vocabulary continuous speech recognition. In *Proc. of ICACCP 1992*, pages I13–I16, San Francisco, USA, Mar 1992.

[33] H. Hermansky. Perceptual linear predictive (PLP) analysis for the speech. *J. Acous. Soc. Am.*, pages 1738–1752, 1990.

[34] H. Hermansky, D. P. W. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. ICASSP 2000*, Turkey, 2000.

[35] H. Hermansky and P. Fousek. Multi-resolution RASTA filtering for tandem-based ASR. In *Proc. of Interspeech 2005*, Lisbon, Portugal, September 2005.

[36] H. Hermansky and P. Jain. Band-independent speech-events categories for TRAP based ASR. In *Proc. Eurospeech 2003*, pages 1013–1016, Geneva, Switzerland, 2003.

[37] H. Hermansky and N. Morgan. RASTA processing of speech. *Trans. on Speech & Audio Processing*, 2(4):578–589, 1994.

[38] H. Hermansky and N. Nabyath. Spectral basis functions from discriminant analysis. In *Proc. of ICSLP 1998*, Sydney, Australia, Dec 1998.

[39] 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*, Paris, France, September 2000.

[40] F. Hönig, G.Stemmer, C. Hacker, and F. Brugnara. Revising perceptual linear prediction (PLP). September 2005.

[41] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[42] M. J. Hunt. A statistical approach to metrics for word and syllable recognition. *J. Acoust Soc. Am.*, 66(S1)(S35(A)), 1979.

[43] S. Ikbal, H. Misra, and H. Bourlard. Phase AutoCorrelation (PAC) derived Robust Speech Features. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-03)*, Hong Kong, China, April 2003.

[44] Anil K. Jain. *Fundamentals of digital image processing.* Prentice Hall, 1988.

[45] P. Jain. *Temporal patterns of frequency-localized features in ASR.* PhD thesis, OGI School of Science and Engineering Oregon Health and Science University, July 2003.

[46] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The ICSI meeting corpus. In *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal processing (ICASSP)*, Hong Kong, 2003.

[47] A. Janin, A. Stolcke, X. Anguera, K. Boakye, O. Cetin, J. Frankel, , and J. Zheng. The ICSI-SRI Spring 2006 meeting recognition system. In *3nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, May 2006.

[48] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[49] S. Kajarekar, B. Yegnanarayana, and H. Hermansky. A study of two dimensional linear discriminants for ASR. In *Proc. of ICASSP 2001*, Salt Lake City, Utah, USA, May 2001.

[50] N. Kanebara, T. Arai, H. Hermansky, and M. Pavel. On the importance of various modulation frequencies for speech recognition. In *Proc. EUROSPEECH 97*, Rhodes, Greece, September 1997.

[51] M. Kleinschmidt and D. Gelbart. Improving word accuracy with Gabor feature extraction. In *Proc. of ICSLP 2002*, Denver, Colorado, USA, 2002.

[52] R. Kompe. Prosody in speech understanding systems. *Lecture Notes in Artificial Intelligence*, 1307, 1997.

[53] S. Kontár. Parallel training of neural networks for speech recognition. In *Proc. 12th International Conference on Soft Computing MENDEL'06*, 2006.

[54] N. Kumar. *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*. PhD thesis, John Hopkins University, Baltimore, 1997.

[55] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9:171–185, 1995.

[56] R.G. Leonard. A database for speaker independent digit recognition. In *ICASSP84*, 1984.

[57] R.P. Lippmann. Speech recognition by machines and humans. *Speech Communication*, 22(1):1–16, 1997.

[58] P. Lockwood and J. Boudy. Experiments with a nonlinear spectral subtractor (NSS), hidden Markov models and the projection, for robust speech recognition in cars. *Speech Communication*, 11:215–228, June 1992.

[59] D. Mansour and B.H. Juang. A family of distortion measures based upon projection operation for robust speech recognition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 36–39, Adelaide, Australia, April 1988.

[60] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence*, pages 374–388, 1976.

[61] G. Miller and P. Nicely. An analysis of perceptual confusions among some english consonants. *The Journal of the Acoustical Society of America*, 27(2):338–352, 1955.

[62] Shang ming Li, Shi-Hao Fang, Jeih weih Hung, and Lin shan Lee. Improved MFCC feature extraction by PCA-optimized filter-bank for speech recognition. In *2001 Automatic Speech and Understanding Workshop*, pages 49–52, Italy, Dec 2001.

[63] H. Misra, H. Bourlard, and V. Tyagi. New entropy based combination rules in HMM/ANN multi-stream ASR. In *Proc. ICASSP 2003*, Hong Kong, China, 2003.

[64] H. Misra, S. Ikbal, H. Bourlard, and H. Hermansky. Spectral entropy based feature for robust ASR. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Canada, May 2004. IDIAP-RR 2003 56.

[65] N. Morgan and H. Bourlard. Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach. *IEEE Signal Processing Magazine*, 12(3):25–42, 1995.

[66] N. Morgan, B. Chen, Q. Zhu, and A. Stolcke. Scaling up: Learning large-scale recognition methods from small scale recognition tasks. In *Proc. Special Workshop in Maui (SWIM)*, 2004.

[67] N. Morgan, B. Y. Chen, Q. Zhu, and A. Stolcke. Trapping conversational speech: Extending TRAP/TANDEM approaches to conversational telephone speech recognition. In *Proc. ICASSP 2004*, Montreal, Canada, May 2004.

[68] P. Motlíček. *Modeling of Spectra and Temporal Trajectories in Speech Processing.* PhD thesis, Brno University of Technology, Faculty of Information Technology, August 2003.

[69] J.P. Openshaw and J.S. Mason. On the limitations of cepstral features in noise. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages II49–II52, Adelaide, Australia, April 1994.

[70] D. Pearce. Enabling new speech driven services for mobile devices: An overview of the ETSI standards activities for distributed speech recognition front-ends. In *Applied Voice Input/Output Society Conference (AVIOS2000)*, San Jose, CA, May 2000.

[71] D. Pearce and H. G. Hirsch. The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *Proc. of ISCLP 2000*, Beijing, China, October 2000.

[72] I. Potamitis, N. Fakotakis, and G. Kokkinakis. Spectral and cepstral projection bases constructed by independent component analysis. In *Proc. of ICSLP 2000*, pages 63–66, Beijing, China, Oct 2000.

[73] Cole R., Fanty M., Noel M., and Lander T. Telephone speech corpus development at CSLU. In *Proc. of ISCLP 1994*, pages 1815–1818, Yokohama, Japan, 1994.

[74] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, February 1989.

[75] D. E. Rumelhart, G. E. Hintont, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(4):533–536, 1986.

[76] P. Schwarz, P. Matějka, and J. Černocký. Towards lower error rates in phoneme recognition. In *Proceedings of 7th International Conference Text, Speech and Dialogue 2004*, page 8, 2004.

[77] P. Schwarz, P. Matějka, and J. Černocký. Hierarchical structures of neural networks for phoneme recognition. In *ICASSP*, Tolouse, France, may 2006.

[78] Sangita R. Sharma. *Multi-stream approach to robust speech recognition.* PhD thesis, Oregon Graduate Institute of Science and Technology, October 1999.

[79] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke. Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, 46(3-4):455 – 472, 2005.

[80] E. Shriberg and A. Stolcke. Prosody modeling for automatic speech recognition and understanding. *Mathematical Foundations of Speech and Language Processing*, 138:105 – 114, 2004.

[81] R. Silipo, S. Greenberg, and T. Arai. Temporal constraints on speech intelligibility as deduced from exceedingly sparse spectral representations. In *Proc. EUROSPEECH'99*, pages 2687–2690, Budapest, Hungary, September 1999.

[82] S. Sivadas. *Tandem Feature Extraction for Automatic Speech Recognition.* PhD thesis, OGI School of Science & Engineering Oregon Health & Science University, November 2004.

[83] A. Stolcke, X. Anguera, K. Boakye, O. Cetin, F. Grézl, A. Janin, A. Mandal, B. Peskin, C. Wooters, and J. Zheng. Further progress in meeting recognition: The ICSI-SRI Spring 2005 speech-to-text evaluation system. In *Proc. NIST MLMI Meeting Recognition Workshop*, Edinburgh, UK, 2005.

[84] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Rao Gadde, M. Plauche, C. Richey, E. Shriberg, K. Sonmez, F. Weng, and J. Zheng. The SRI March 2000 hub-5 conversational speech transcription system. In *Proc. NIST Speech Transcription Workshop*, College Park, MD, 2000.

[85] P. Svojanovský. Band-independent classifiers in TRAP-TANDEM ASR system. In *Proc. of SPECOM 2005*, pages 769–772, Patras, Greece, Oct 2005.

[86] V. Tyagi and C. Wellekens. Fepstrum representation of speech signal. In *Proc of IEEE ASRU*, pages 44–49, San Juan, Puerto Rico, Dec 2005.

[87] F. Valente and H. Hermansky. Discriminant linear processing of time-frequency plane. Technical Report 06-20, IDIAP Research Institute, Martigny, Switzerland, Apr 2006.

[88] S. van Vuuren and H. Hermansky. Data-driven design of RASTA-like filters. In *Proc. of Eurospeech 1997*, Rhodos, Greece, Sep 1997.

[89] A.P. Varga and R.K. Moore. Hidden Markov model decomposition of speech and noise. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 845–848, Adelaide, Australia, 1990.

[90] J. Černocký. Temporal processing for feature extraction in speech recognition. Habilitation thesis, Brno University of Technology, Faculty of Information Technology, October 2002.

[91] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13:260–269, Apr 1967.

[92] R. M. Warren and J. A. Bashford. Intelligibility of 1/3-octave speech: Greater contribution of frequencies outside than inside the nominal passband. *The Journal of the Acoustical Society of America*, 106(5):47–52, Nov 1999.

[93] R. M. Warren, K. R. Riener, Jr. J. A. Bashford, and B. S. Brubaker. Spectral redundancy: Intelligibility of sentences heard through narrow spectral slits. *Perception and Psychophysics*, 57(2):175–182, 1995.

[94] P. J. Werbos. *The Roots of Backpropagation*. Wiley-Interscience, New York, 1994.

[95] D. Willett, C. Neukirchen, and G. Rigol. DUCODER-the duisburg university LVCSR stack decoder. In *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal processing (ICASSP)*, Istanbul, 2000.

[96] H. H. Yang, S. Sharma, S. van Vuuren, and H. Hermansky. Relevance of time-frequency features for phonetic and speaker-channel classification. *Speech Communication*, 31(1):35–50, Aug 2000.

[97] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland. *The HTK book*. Entropics Cambridge Research Lab., Cambridge, UK, 2002.

[98] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke. On using MLP features in LVCSR. In *Proc. ICSLP 2004*, Jeju Island, KR, Oct 2004.

[99] Q. Zhu, A. Stolcke, B. Chen, and N. Morgan. Using MLP features in SRI's conversational speech recognition system. In *Proc. INTERSPEECH 2005*, Lisbon, Portugal, September 2005.

Frequent abbreviations

ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
CRBE	CRitical Band Energy
CRBS	CRitical Band Spectrogram
CTS	Conversational Telephone Speech
CV	Cross Validation
CVFA	Cross Validation Frame Accuracy
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DHT	Discrete Hadamard Transform
FA	Frame Accuracy
FA	Frequency Averaging (modifying operator)
FD	Frequency Differentiation (modifying operator)
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
HATS	Hidden Activation TRAPS
HLDA	Heteroscedastic Linear Discriminant Analysis
HMM	Hidden Markov Model
HTK	Hidden Markov model ToolKit
ICA	Independent Component Analysis
KLT	Karhunen-Loève Transform
LDA	Linear Discriminant Analysis
LM	Language Model
LP	Linear Prediction
LPC	Linear Prediction Coefficients
MCRBS	Modified CRitical Band Spectrogram
MFCC	Mel-Frequency Cepstral Coefficients
MLLR	Maximum Likelihood Linear Regression
MLP	Multi-Layer Perceptron
MO	Modifying Operator
MRASTA	Multi-resolution RASTA
MTRAP	Modified TemopoRAl Pattern
PCA	Principal Component Analysis
PLP	Perceptual Linear Predictive (coefficients)
RASTA	Relative Spectra
SNR	Signal to Noise Ratio

TA	Time Averaging (modifying operator)
TD	Time Differentiation (modifying operator)
TMLP	Tonotopic Multi-Layered Perceptron
TRAP	TempoRAl Pattern
WE	Word External (models)
WER	Word Error Rate
WI	Word Internal (models)

List of appendices

Appendix A

Phoneme set

The following table gives the IPA, Worldbet and OGIbet notation of English phonemes with example words.

IPA, Worldbet, and OGIbet English Broad Phonetic Labels

Center for Spoken Language Understanding – Oregon Graduate Institute of Science & Technology

IPA	Worldbet	OGIbet	Example	Category
iː	i:	iy	beet	
ɪ	I	ih	bit	Front
ɛ	E	eh	bet	Vowels
æ	@	ae	bat	
ɨ	I_x	ix	roses	
ʉ	u_x	ux	suit	Central
ə	&	ax	above	Vowels
ɚ	&0		to go	
ɐ	5		pot	(British)
u	u	uw	boot	
ʊ	U	uh	book	Back
ʌ	^	ah	above	Vowels
ɔ	>	ao	caught	
ɑ	A	aa	father	
ɝ	3r	er	bird	Retro-
ɚ	&r	axr	butter	flexes
ei	ei	ey	bay	
aɪ	aI	ay	bye	Diph-
ɔi	>i	oy	boy	thongs
iʊ	iU		few	
aʊ	aU	aw	about	
oʊ	oU	ow	boat	
iə	i&		here	(British)
eə	e&		there	(British)
uə	u&		poor	(British)
p^h	ph	p	pan	Voiceless
t^h	th	t	tan	Plosives
k^h	kh	k	can	
b	b	b	ban	Voiced
d	d	d	dan	Plosives
g	g	g	gander	
m	m	m	me	
n	n	n	knee	Nasals
ŋ	N	ng	sing	
ɾ$_t$	th_(	dx	writer	Flaps
ɾ$_d$	d_(	dx	rider	
f	f	f	fine	
θ	T	th	thigh	Voiceless
s	s	s	sign	Fricatives
ʃ	S	sh	assure	
h	h	hh	hope	
v	v	v	vine	
ð	D	dh	thy	Voiced
z	z	z	resign	Fricatives
ʒ	Z	zh	azure	
ʧ	tS	ch	church	Affricates
ʤ	dZ	jh	judge	
l	l	l	lent	Glides
ɻ	9r	r	rent	
j	j	y	yes	(approxi-
w	w	w	went	mants)
m̩	m=	em	bottom	
n̩	n=	en	button	Syllabics
ŋ̩	N=	eng		
l̩	l=	el	bottle	

IPA	Worldbet	OGIbet	Example	Category
	pc	pcl	_pan	Voiceless
	tc	tcl	_tan	Plosive
	kc	kcl	_can	Closures
	bc	bcl	_ban	Voiced
	dc	dcl	_dan	Plosive
	gc	gcl	_gander	Closures
	tSc	chcl	_church	Affricate
	dZc	jhcl	_judge	Closures
	+	.epi	epinthetic closure	

IPA	Worldbet	OGIbet	Type of Diacritic
t^h	_h	-h	aspirated
	_x		centralized
t̪ d̪	_[		dental
	_(		flapped (consonant)
	_F		fricated stop
	_?*	q	glottal onset
ʔ	_?	-q	glottalized
d'	_l		lateral release
iː	_:	-el	lengthened
d^n	_n		nasal release
ẽ	_~	-n	nasalized
	_NL	.nitl	not in the language
t^j	_j		palatalized
ɚ	_r	-r	retroflexion
ɹ	_i		less rounded
ɻ	_w		more rounded
ɬ	_=		syllabicity
s̬	_v		voiced
n̥ d̥	_0		voiceless
	_*	-	waveform cut off
Worldbet, as modified at OGI			
	_fp	-fp	filled pause
	_ln	-ln	line noise corruption
	_bn		background noise

Worldbet	OGIbet	Non Speech Sound Item
.bn	.bn	background noise
.br	.br	breath noise
.cough	.cough	cough
.ct	.ct	clear throat
.laugh	.laugh	laugh
.ln	.ln	lin noise
.ls	.ls	lip smack
.ns	.ns	human, not speech
.sneeze	.sneeze	sneeze
.tc	.tc	tongue click
Worldbet, as modified at OGI		
.beep	.beep	beep
.burp	.burp	burp
.fp	.fp	filled pause
.pau	.pau	pause or silence
.sniff	.sniff	sniff
.uu	.unk	unintelligible speech
.vs	.vs	squeak, voice crack
.glot	glot	glottalization

Bibliographical Note

František was born on March 16, 1977 in Šternberk, Czech Republic. He received his Master's degree in Electrical Engineering from the Brno University of Technology in June 2000. The topic of his diploma project was the design of Fire's channel coders with adjustable protection capability.

He joined the Laboratory of Signal Processing at Brno University of Technology as a Ph.D. student in September 2000. There he started to work on feature extraction for speech recognition. He was on an internship in Anthropic Signal Processing group at Oregon Graduate Institute in Portland, USA, in May 2001. Under a guidance of Prof. Hynek Hermansky, he worked on projects SPINE, AURORA2 and EARS. He returned to Brno University in March 2003 and worked on M4 project. In January 2004, he was on internship at IDIAP in Martigny, Switzerland, where he rejoined the EARS project. In October 2004, he was accepted for AMI Training Programme at ICSI, Berkeley, USA and IDIAP, Martigny, Switzerland. After his return to Brno University in September 2005, he continued to work on AMI project.

During his graduate studies, František has authored and co-authored several conference papers presented on international events. He taught lectures and exercizes in signal and speech related courses and led several Bc. and Ms. thesis. He is member of IEEE and "Český svaz včelařů".