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

HYBRIDNÍ ROZPOZNÁVAČ IZOLOVANÝCH SLOV

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE KAREL VESELÝ
AUTHOR

BRNO 2007

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

HYBRIDNÍ ROZPOZNÁVAČ IZOLOVANÝCH SLOV

HYBRID RECOGNIZER OF ISOLATED WORDS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

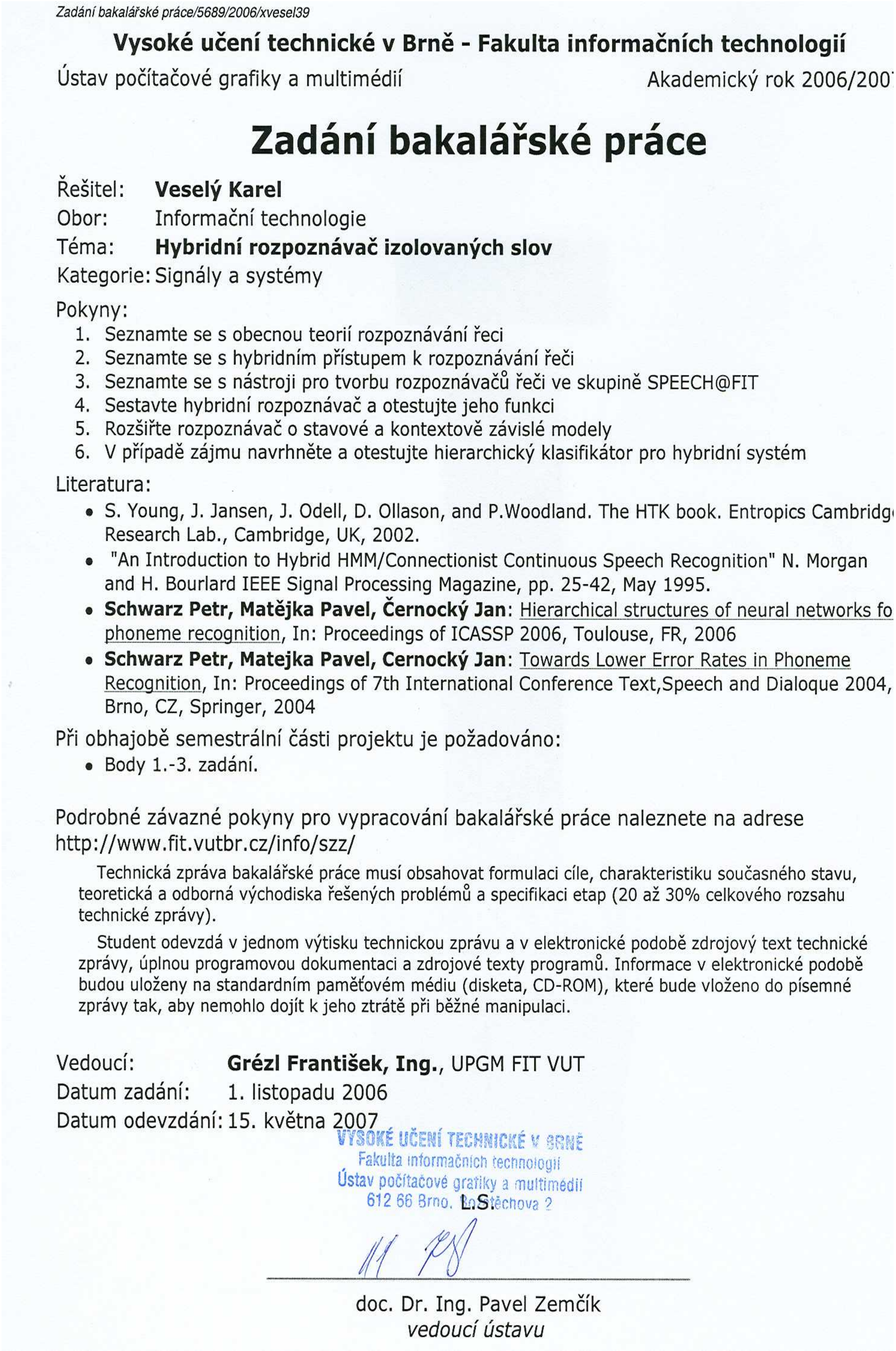
AUTOR PRÁCE
AUTHOR

KAREL VESELÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. FRANTIŠEK GRÉZL

BRNO 2007



Zadání bakalářské práce/5689/2006/xvesel39

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2006/200

Zadání bakalářské práce

Řešitel: **Veselý Karel**

Obor: Informační technologie

Téma: **Hybridní rozpoznávač izolovaných slov**

Kategorie: Signály a systémy

Pokyny:

1. Seznamte se s obecnou teorií rozpoznávání řeci
2. Seznamte se s hybridním přístupem k rozpoznávání řeči
3. Seznamte se s nástroji pro tvorbu rozpoznávačů řeči ve skupině SPEECH@FIT
4. Sestavte hybridní rozpoznávač a otestujte jeho funkci
5. Rozšiřte rozpoznávač o stavové a kontextově závislé modely
6. V případě zájmu navrhněte a otestujte hierarchický klasifikátor pro hybridní systém

Literatura:

- S. Young, J. Jansen, J. Odell, D. Ollason, and P.Woodland. The HTK book. Entropics Cambridg Research Lab., Cambridge, UK, 2002.
- "An Introduction to Hybrid HMM/Connectionist Continuous Speech Recognition" N. Morgan and H. Bourlard IEEE Signal Processing Magazine, pp. 25-42, May 1995.
- **Schwarz Petr, Matějka Pavel, Černocký Jan**: Hierarchical structures of neural networks fo phoneme recognition, In: Proceedings of ICASSP 2006, Toulouse, FR, 2006
- **Schwarz Petr, Matejka Pavel, Cernocký Jan**: Towards Lower Error Rates in Phoneme Recognition, In: Proceedings of 7th International Conference Text,Speech and Dialoque 2004, Brno, CZ, Springer, 2004

Při obhajobě semestrální části projektu je požadováno:

- Body 1.-3. zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Grézl František, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Pavel Zemčík
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Karel Veselý

Bytem: Dukelských bojovníku 2820/125, Znojmo, 671 81

Narozen/a (datum a místo): 28.9.1984 ve Znojmě

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií

se sídlem: Božetěchova 2, 612 66, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

....

(dále jen „nabyvatel“)

Čl. 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:	Hybrid recognizer of isolated words
Vedoucí/ školitel VŠKP:	Ing. Grézl František
Ústav:	Ústav počítačové grafiky a multimédií
Datum obhajoby VŠKP:	

VŠKP odevzdal autor nabyvateli v[*]:

□ tištěné formě – počet exemplářů ………………..

□ elektronické formě – počet exemplářů ………………..

[*] hodící se zaškrtněte

1. 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.
2. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
3. Autor potvrzuje, že listinná a elektronická verze díla je identická.

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: …………………………………….

………………………………………..	…………………………………………
Nabyvatel	Autor

Abstrakt

Rozpozávač izolovaných slov nezávislý na mluvčím má mnoho praktických použití. Například bude umožňovat ovládat hlasem různé domácí přístroje příští generace které budou komunikovat s PC. Ještě zajímavější je možnost jej vestavět do jakékoli aplikace nebo dokonce do operačního systému a rozšířit tak uživatelské rozhranní o nový prvek, hlasové ovládání. Dá se využít k ovládání pomocí klíčových slov, reakcí může být spuštění aplikace nebo jakákoli jiná specifická akce. Nejzajímavější možnost využití rozpoznávače izolovaných slov je v elektronických slovnících. Novým rysem slovníků příští generace by mohlo být hlasové vyhledávání slov. Velmi užitečná je možnost získat na výstupu seznam slov sežazený podle pravděpodobnosti vyslovení. Tento rys umožňuje uživateli jednoduše zjistit podobná slova a naučit se je lépe rozlišovat.

Klíčová slova

rozpoznávání řeči, izolovaná slova, hybridní rozpoznávač, neuronová síť, foném, FBANK, VTLN, N-best, elektronický slovník

Abstract

The speaker independent isolated words recignizer has various practical applications. For example it can be used to control home gadgets by PC. Even more interesting is possibility that it can be built in the user interface of any application or even into operating system to perform command based control such as invocation of applications, or execution of any other specific action. The most remarkable application of isolated recognition is in electronical dictionaries. A voice controlled word lookup could be new feature of the next generation dictionaries. Very useful is the ability to ouptut ordered list of the most likely words, which gives the user ability to learn and distinguish similar words.

Keywords

speech recognition, isolated words, hybrid recognizer, neural network, phoneme, FBANK, feature extraction, VTLN, frequency warping, N-best, multi layer perceptron, electronic dictionary

Citace

Karel Veselý: Hybridní rozpoznávač izolovaných slov, bakalářská práce, Brno, FIT VUT v Brně, 2007

Hybridní rozpoznávač izolovaných slov

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Františka Grézla. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

. . . .
Karel Veselý
14. května 2007

Poděkování

Tento projekt byl částečně podporován Grantovou agenturou České republiky v rámci projektu č. 102/05/0278 dále pak Ministerstvem školství mládeže a tělovýchovy v rámci projektu č. MSM0021630528. Použitý hardware byl částečně poskytnut organizací CESNET v rámci projektů č. 119/2004, č. 162/2005 a č. 201/2006.

Contents

Chapter 1

Introduction

The very good application of isolated word recognition is in the *electronic dictionaries*. It can be easily used to look up words and maybe phrases. The key feature is the N-best recognition which outputs ordered set of N words which are the most likely. The probability, that the correct word is within this set, is much higher than just recognition of the most likely word. Another advantage of N-best recognition is that similar words will be output. In case of wrong pronunciation the intended word will be within the output. And last but fundamental use case is when the user is sure about the word pronunciation and wants to know the orthography.

A typical isolated word recognizer is based on statistical pattern recognition and its goal is to find the best word given a set of input patterns (observations) and modeling parameters. Let $\mathbf{X} = x_1, x_2, \ldots, x_N$ be a sequence of N observation vectors, feature vectors. Let $\mathbf{W}$ be a word. The ASR system output is such word $\overline{\mathbf{W}}$ which maximizes equation:

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

Where Θ is a set of all modeling parameters. Instead of building overall model $P(\mathbf{W}|\mathbf{X}, \Theta)$ we can factor this into smaller models. First, we can split the words into a distinct sounds. The phonemes[1] are 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 each word $\mathbf{W}$. Thus the word $\mathbf{W}$ in Eq. (1.1) can be replaced by all possible sequences of phones $\mathbf{Q}$ which together form word $\mathbf{W}$:

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

By using Bayes rule we can further obtain:

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

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

[1] By the Wikipedia definition the phoneme is the conception of speech sound in the most neutral form possible. It is also the smallest sound unit, which distinguishes between two different words. Cardinality of phonemes differs per *phonemic systems*, but there is often around forty-five of them for English.

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

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

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

The three probability models in Eq. (1.5) are:

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

Pronunciation model $P(\mathbf{Q}|\mathbf{W},\Theta_{PM})$ which tells us how probable is a sequence of phonemes of given word. The pronunciation model is also called “pronunciation dictionary” or only “dictionary”.

Language model $P(\mathbf{W}|\Theta_{LM})$ which is not considered in isolated word recognition and is replaced by simple parallel grammar network.

The direct estimation of the joint conditional probability $P(x_1, x_2, \ldots, x_N|\mathbf{Q},\Theta_{AM})$ of the spoken word is not practicable. A parametric model of word production such as a Markov model is assumed.

Speech recognition systems generally assume that the speech signal is a realisation of some message encoded as a sequence of one or more symbols (eg. words, phonemes). The *automatic speech recognition* (ASR) systems are designed to extract these symbols from the waveforms.

In the real system the speech signal is first preprocessed and segmented into equally distanced frames with the same width. Typical frame width is 25ms with 10ms shift. The speech frames can be regarded as stationary. The small piece of speech signal is then transformed into *feature vector*, which captures the relevant characteristic of the speech frame for further processing. This process is called **Feature extraction** or Parametrisation of speech.

The role of the recogniser is to perform a mapping between sequences of feature vectors and the wanted underlying symbol sequences. Two problems make this very difficult. First, the mapping from symbols to speech is not one-to-one since different symbols can give similar sounds. Furthermore, there are large variations in the speech waveform of the same symbol due to speaker variability, mood, environment, etc. Second, the boundaries between symbols cannot be identified explicitly from the speech waveform. Hence, it is not possible to treat the speech waveform as a sequence of concatenated static patterns.

The hybrid recognition system is a system which estimates phoneme class probabilities by *Artificial Neural Network* (ANN), and decodes them into words by *Hidden Markov models* (HMMs), according the schema in the figure 1.1.

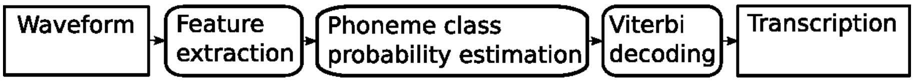


Figure 1.1: Recognition process

On the feature extraction level the speech signal is segmented into frames, transformed by FFT and integrated by an auditory motivated filter bank. An equal loudness normalisation within the bands can be performed. The feature vectors are not decorrelated because it is not necessary for hybrid recognition.

Then the phoneme recognition is performed. On the phoneme classification level the feature vectors are mapped to vectors of phoneme class membership probability. For this purpose is used a statistical model. This model is Multi-Layer Perceptron in case of hybrid recognition. MLP is a specific type of Artificial Neural Network.

On the decoding level the vectors of estimated probabilities are decoded to words by a standard Viterbi algorithm[1]. The underlying words are decoded from a recognition network which is constructed from a word-level grammar network[2] , the word to phonemes dictionary and a phoneme Hidden Markov Models (HMM) set.

As the likelihoods of the HMM states are used the phoneme class estimator outputs.

[2] The grammar network is substitute to language model in case of hybrid recognition.

Chapter 2

System description

In this chapter the hybrid isolated word recognizer is described into detail. The following topics are the test data set, models, feature extraction, feature normalization, phoneme classification and decoding with the grammar, dictionary and HMMs. The N-best recognition and context dependent phoneme modeling are also introduced.

For the construction of speech recognizer were used HTK and STK toolkit. The used commands are explained and included to show a simplified overview how to build a hybrid recognizer. For more information about these toolkits see the following links:

HTK: http://htk.eng.cam.ac.uk/

STK: http://www.fit.vutbr.cz/research/groups/speech/index.php?id=stk

A good knowledge of perl scripting is a big advance for a successful construction of recognizer with optimisation options, batch processing and evaluation of large test speech database.

2.1 Test data set

All the experiments were performed over a set of word level transcribed speech records. The speech database can be characterized as follows:

- 13 sets of speech from 12 speakers
- The sets were recorded on different hardware in different locations.
- Native and non-native speakers
- Each set contains 995 words of the same.
- The sets were recorded in normal office environment with background noise.

The speech data were supplied by our partners of project. For the subjective characteristics of record sets see the table 2.1. The speech was supplied in CD quality 44KHz sampling, 16 bits per sample. For purpose of recognition it had to be downsampled to 16KHz and 8KHz. It has been done by sox utility using polyphase filtering.

Set	Quality description	Sex
01a	overexcited microphone, power-line hum, blowing into microphone, background noises	female
01b	power-line hum, background noises	female
02a	power-line hum, few words cut incorrectly	female
03a	power-line hum, weak signal, no background noises	female
04a	hum, relatively good SNR	female
05a	good SNR, some hiss	male
06a	overexcited microphone, few records proper	female
07a	perfect SNR, good set	male
08a	weak hum, weak bg. noises	female
09a	medium noise (middles, like PC-fan), quiet speech	male
10a	medium noise (middles, like PC-fan)	male
11a	power-line hum, background noises	female
12a	strong noise (low-middles, like loud PC-fan)	male

Table 2.1: Speech sets characteristics

2.2 Models

For the experiments were used several phoneme class estimators of different design. All of them were provided by the group SPEECH@FIT as completely trained system components.

The supplied models are:

A - `MLP4_both_31crbe15_dct16_240_829_829_135_x4.sfcat_log`

B - `SCnoVTLN_Merg_both_crbe23_dct11_SNN3_253_1500_135.sfcat`

C - `SC_Merg_both_crbe23_dct11_SNN3_253_1500_135_cont_merg.sfcat`

D - `SC_Merg_both_crbe23cmncvn_dct11_SNN3_253_1500_135_cont_merg.sfcat`

E - `MLP4_both_31crbe23cmncvn_dct16_368_639_639_555_cont.sfcat`

All the models works with FBANK features, each model was trained on specific type of features thus the same type of features is necessary for recognition. The mismatch of the feature type leads to failure of the system.

The input features to model A are normalized, they have 15 critical bands.
The input features to model B have 23 critical bands.
The input features to model C have 23 critical bands and VTLN.
The input features to model D are normalized, they have 23 critical bands and VTLN.
The input features to model E are normalized, they have 23 critical bands and VTLN.

The models preprocess of the features is based on assumption, that the speech can be threated in critical bands separately during the early stages of processing. The band-specific knowledge can be merged later. This assumption is supported by many psycho-acoustic experiments[5], showing the ability of humans to correctly recognize speech even from narrow frequency band.

All the models work with 31 frames[1] long context, the actual frame plus 15 frames to both directions. The temporal critical band trajectories are first processed by *Hamming window* to put less weight on farther frames from actual frame. Afterwards they are post-processed by DCT (*Discrete Cosine Transform*), linear projection, which is used to reduce dimensionality from 31 to 16. The DCT represents the band independent processing and prevents ANN from hamming window compensation. There is usually no loss of information and no degradation in phoneme error rate caused by this projection[3].

The DCT parameters are globally mean and variance normalized and put into *Multi-Layer Perceptron*, which produces the class posterior probabilities.

The classes are based on phoneme states. For models A–D the context independent phoneme states are used as classes. There are 45 phonemes in our system, each of them is modelled by 3 states. This gives us 135 outputs. The classes for E are based on context dependent phonemes. We used 555 classes, which were made by *tying*.

The detail inner structure of model A can be viewed in the figure 2.1, see the figure 2.2 for the less detailed view on the whole recognition system.

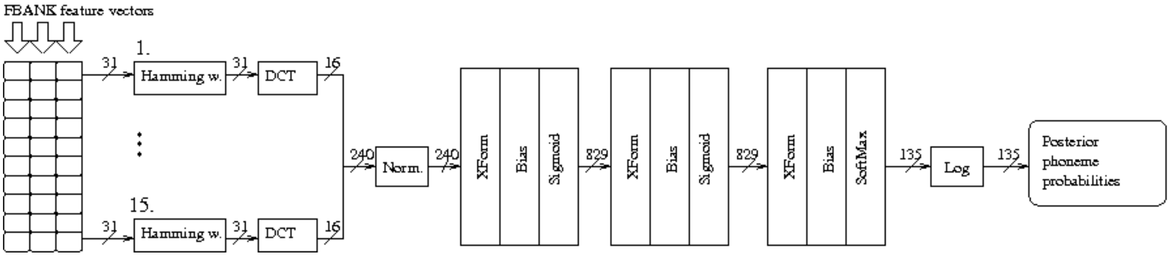


Figure 2.1: Anatomy of phoneme estimator with 4 layer MLP

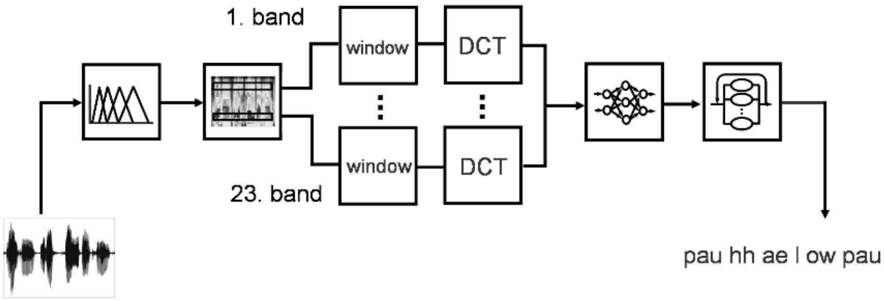


Figure 2.2: Hybrid recognizer

[1] 31 frames context is 325ms long for frame width 25ms and shift 10ms

The models B–D have different design, which contains split time context. They all contain three 3-Layer MLPs connected into hierarchical structure. Two of the three MLPs are separately trained phoneme estimators, one for the left time context and the other for the right time context. The third MLP is used to merge the final outputs. The scheme of this conformation is depicted in the figure 2.3.

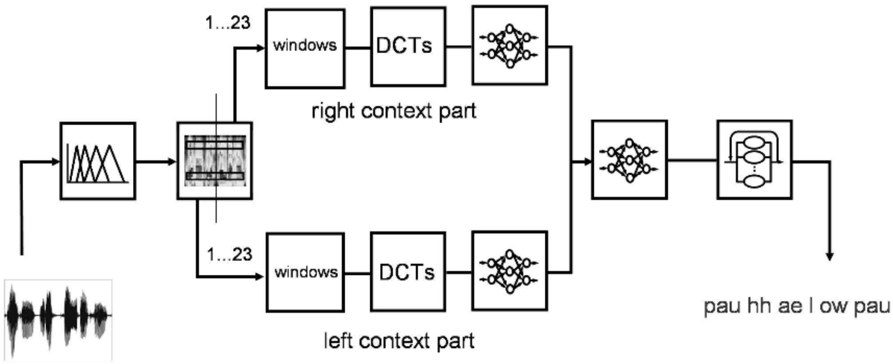


Figure 2.3: system with split left and right temporal context

The MLP layers count and number of features[2] are good metrics to compare the models, see the table 2.2.

Model	**Layer count**	**Feature count**
A	4 layers	999909
B–D	three 3-layer MLPs	1776405 (2 x 583635 - time contexts, 609135 - merger)
E	4 layers	999951

Table 2.2: Model size comparison

The model A was trained on 100 hours of 8KHz telephone speech. The models B–E were trained on 30 hours of 16KHz office environment meeting speech.

2.3 Feature extraction

For the system was used **FBANK** parametrisation which are *CRitical Band Energies* (CRBE) obtained.

Initially each speech frame is transformed by FFT, the spectral magnitudes are integrated by the auditory motivated filter bank, which consists of N triangles equally distanced on Mel-scale.

The humans ear resolve frequencies non-linearly, thus the frequencies are transformed by equation (2.1) into Mel-frequency scale, in which the frequencies are perceived as linear. This transform is used according to the assumption that the machines are able to process the speech in a similar manner like humans.

[2] The feature number of model is the count of coeffitients which had to be trained.

$$\text{Mel}(f) = 2595 \log_{10}(1 + \frac{f}{700}) \tag{2.1}$$

The sample shape of Mel-filter bank which is used for integration can be seen in figure 2.4. Each triangle represents one critical band.

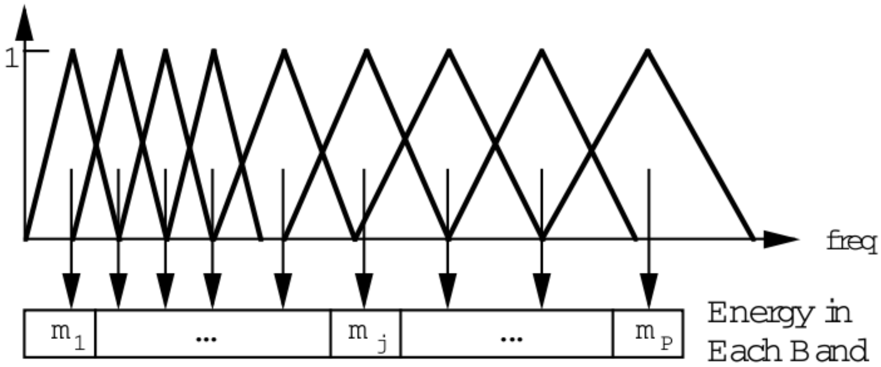


Figure 2.4: Mel-scale filter bank

It is possible to perform some optimisations on the feature extraction level, such as VTLN (see chapter 3) and feature normalisation (chapter 2.4).

In the figure 2.5 can be seen a Mel-scale spectrogram, the FBANK speech representation. The X axis is the number of speech frame, the Y axis the number of the critical band. The dark shape in the middle is the word itself, the light left and right margins represent silence in the signal. The record had strong hum in the critical band number 1. See the dark horizontal line within the utterance independent on presence of the speech.

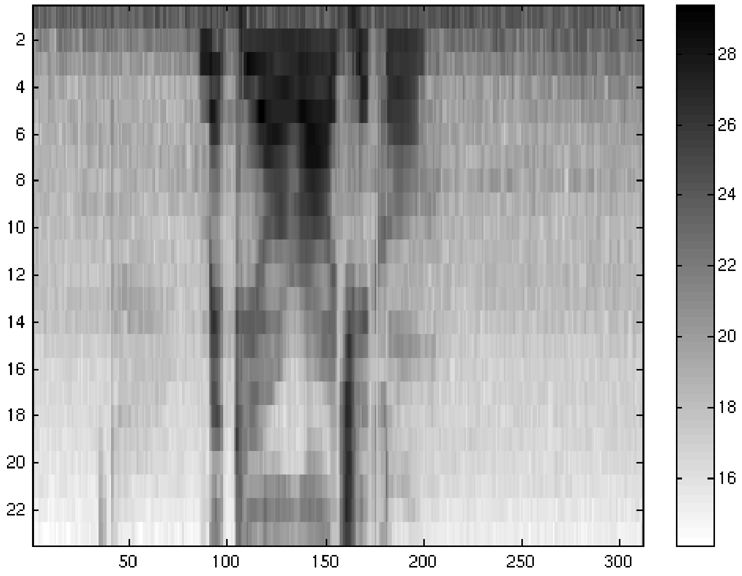


Figure 2.5: FBANK representation of word "meteorological"

Different lengths of feature vectors were used for different sampling rates. One critical band is represented one coefficient in the feature vector. For 8KHz waveforms were used 15 critical bands and for 16KHz 23 critical bands.

In our case the feature extraction is done by the HCopy HTK utility. It is used by the following command:

```
HCopy -S script_file -C config_file
```

The HCopy utility is able to either perform data conversion to parametrised form or simply copy the data files. To perform feature extraction two things are necessary: a config file, which contains parameters such as:

- source and target data format
- window width and shift
- number of critical bands
- whether to use hamming window on speech windows
- whether to compute from signal utterance or its power
- whether to use little-endian or big-endian encoding
...

The other necessity is to prepare a script file, which means in HTK terminology a file that contains a list of files to be processed. For HCopy it must have two columns, in the first are original files in the second the target filenames. It is very useful to write a script which creates the HTK script files, as the parametrisation is a very frequent task. HCopy doesn't create target directories, they must be prepared before the HCopy invocation.

2.4 Mean-variance feature normalization

The mean and variance feature normalisation is be performed if the used model needs it.

The normalisation is per speaker and is done for each critical band separately from all the set files. It improves the statistical characteristics and compensates long-term cepstral effects such as the microphone and signal channel influence.

For the feature normalisation is used HCompV HTK tool.

2.5 Phoneme class probability estimation

Now when the speech data are in suitable representation, the phoneme class estimation can be performed. Each feature vector belongs to all of the phoneme classes with certain likelihoods. These likelihoods are estimated by a statistical model.

The hybrid recognition uses the model based on *Multi-Layer Perceptrons* (MLPs), unlike the conventional approach which uses *Gaussian Mixture Models* (GMMs).

- For the experiments with phoneme estimator A[3] were used 8KHz speech records.
- Split time context estimators B–D were used in the experiments with 16KHz speech records.

[3] The model labels are introduced in chapter 2.2, page 6.

- The 16KHz speech records were used for the experimental system with the context dependent phoneme estimator E.

The estimators A–D are producing 135 long vectors of phoneme state emission probabilities, which can be simply visualised. The phoneme state probability matrix of word ”meteorological” is shown in the figure 2.6.

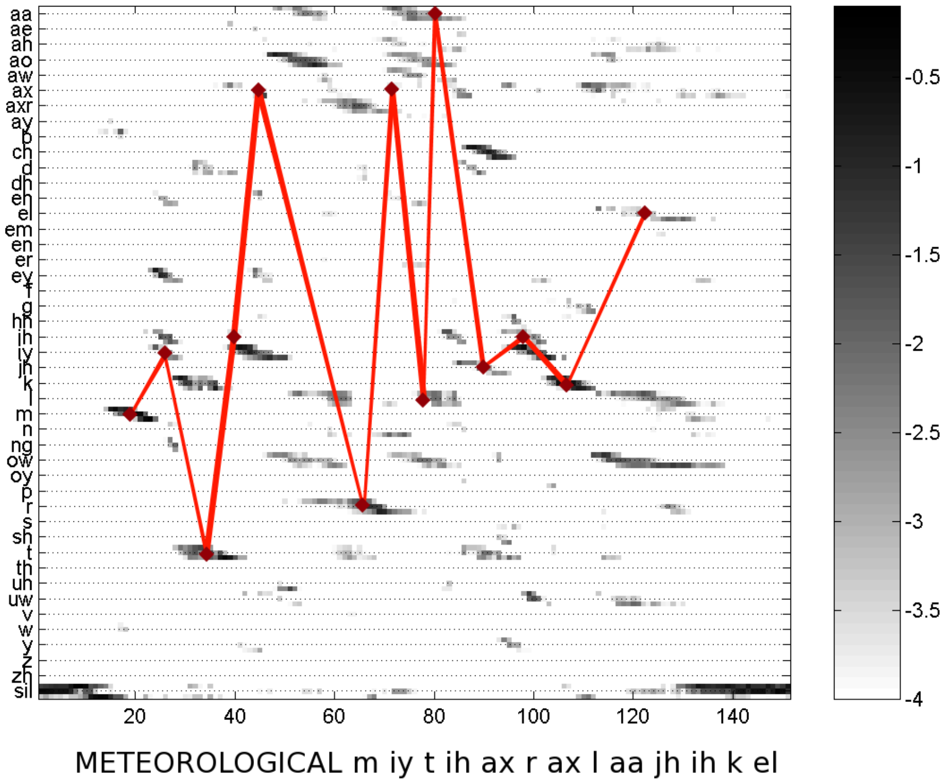


Figure 2.6: phoneme state log-probabilities of word ”meteorological”

The X axis is the number of the speech frame, on the Y axis are the phoneme states. The probabilities are depicted in logarithmic scale. At the bottom corners of the graph are two significant horizontal black lines which represents the silence models. All the black lines in the central region are possible phonemes.

In our system the SFeaCat STK tool is used to perform the phoneme class estimation. It is executed by the following command.

```
SFeaCat --startfrmext=15 --endfrmext=15 -y prob -l ./phn_probs \
-H MLP4_both_31crbe15_dct16_240_829_829_135_x4.sfcat_log \
-S script_file
```

The meaning of the parameters is following:

`--startfrmext=15`	Context initialisation, the phoneme classifier works with 31 frames long context, (actual frame, 15 on left + 15 on right)
`--endfrmext=15`	
`-y prob`	extension of output files
`-l ./phn_probs`	output directory
`-H MLP4_...`	HMM macro file, the transform definition
`-S script_file`	script file, list of feature files

This operation is quite time demanding, most of the time is consumed by the matrix multiplications. The multiplications for 4 layer MLP from the figure 2.1 would be: [240,1]x[829,240], [829,1]x[829,829] and [829,1]x[135,829].

2.6 Decoding

The previous chapter discussed the phoneme class probabilities estimation, this chapter deals how to "read words" from the phoneme state emission probabilities.

Very simplified, the decoding process finds the through-pass of the phoneme class probability matrix which is the most likely.

Decoding in HTK[1] is controlled by a *recognition network* compiled from word-level network, a dictionary and a set of Hidden Markov Models (HMMs). The recognition network consists of a set of nodes connected by arcs. Each node is either a HMM model instance or a word-end. Each model node is itself a network consisting of states connected by arcs. Thus, once fully compiled, a recognition network ultimately consists of HMM states connected by arcs. However, it can be viewed at three different levels: word, model and state.

The model-level network is generated from word-level by expanding the words into phoneme strings according to the dictionary. The state-level will be received by expanding phonemes to their states by related HMMs.

The job of the decoder to assign the likelihood to each path through the expanded network, and choose one or more words according to the likelihoods. The paths are evaluated using a *Token Passing algorithm*[1] which is an alternative formulation of Viterbi algorithm.

The decoder can be used also in different mode, it can perform *forced alignment*[1]. In case of FA, the recognition network contains the correct word only. Usually FA is used for optimization of word likelihoods during training and for VTLN coefficient estimation. More details about VTLN are in the chapter 3.

2.6.1 Dictionary

The pronunciation model is represented by dictionary, in which words are transcribed as concatenation of the phonemes. In the dictionary might be more pronunciation possibilities.

Our dictionary contains 1000 words, each word has one pronunciation option. See the examples from the used dictionary in the table 2.3.

The pronunciation model is usually made by the linguist who has to listen to the spoken terms and decide on the phoneme transcription. The quality of model affects recognition of individual words.

The used dictionary was supplied by our partners in industry. The company uses different set of phonemes thus the verified conversion had to be performed.

Word	Phoneme transcription
ORIENTATION	ao r r ih eh n t ey sh ax n
RIOT	r ay ax t
POTENTIAL	p ax t eh n sh ax l
WINDSCREEN	w ih n d s k r iy n
SENTIMENT	s eh n t ih m ax n t

Table 2.3: Dictionary example

2.6.2 Word-level grammar network

A very simple word-level grammar network is used for isolated word recognition. The words are organized in parallel, the surrounding silence models are connected before and after all the words. The scheme in figure 2.7 illustrates the grammar network shape.

Technically the word-level grammar network is an orientated graph. It is stored in *standard lattice format* (SLF)[1]. The format contains section for nodes (words) and section for arcs (transitions).

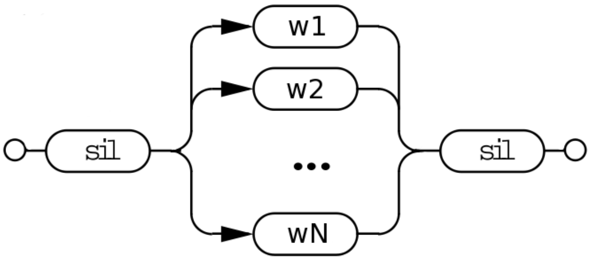


Figure 2.7: Word-level grammar network

2.6.3 HMM models

The HMM models represent the adapter between the phoneme class probabilities and the recognition network. The set of HMMs maps the phoneme classifier output to the HMM states. The mapping is illustrated in the figure 2.8.

For context independent phonemes the *Master Model File* (MMF) contains 45 HMM models, one per phoneme. The MMF file stores the HMM definitions written in *HMM definition language*[1]. The sample HMM definition will be explained on the example in the following lines:

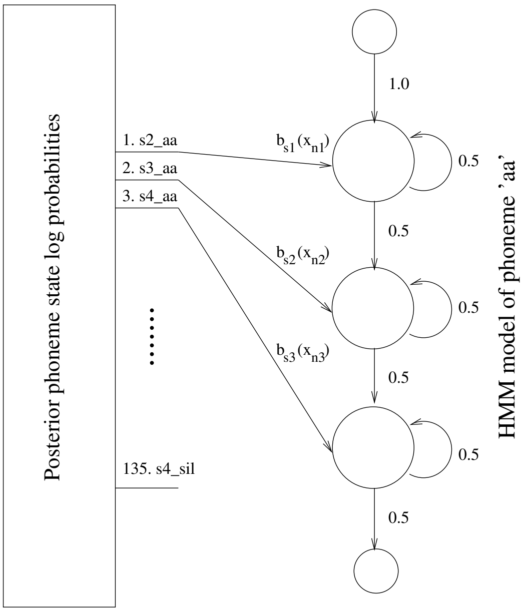


Figure 2.8: Mapping phoneme classifier output into HMM models

```
 1 |  o <VecSize> 135 <PDFObsVec>
 2 |
 3 |  h ’’a’’
 4 | <BEGINHMM>
 5 | <NUMSTATES> 5
 6 | <STATE> 2 <ObsCoef> 1
 7 | <STATE> 3 <ObsCoef> 2
 8 | <STATE> 4 <ObsCoef> 3
 9 | <TRANSP> 5
10 | 0.000000e+00 1.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
11 | 0.000000e+00 5.000000e-01 5.000000e-01 0.000000e+00 0.000000e+00
12 | 0.000000e+00 0.000000e+00 5.000000e-01 5.000000e-01 0.000000e+00
13 | 0.000000e+00 0.000000e+00 0.000000e+00 5.000000e-01 5.000000e-01
14 | 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
15 | <ENDHMM>
```

Line 1 Specification of the data input format, the global variable `VecSize` is set. The length of observation vector is 135, the data format is `PDFObsVec`.

Line 3 Label of HMM definition, label of certain context independent phoneme.

Line 4 Start of HMM definition.

Line 5 Number of states in the HMM. There are five states, although the phoneme is modelled by three states. In HTK are two outer states added to every model for easy binding of successive models.

Lines 6–8 The states are bound with the observation vector cells.

Line 9 Transition matrix header.

Lines 10–14 Transition matrix specification, each line refers to one state. In the line 11, which refers to the second state, is the probability to stay in the state `5.000000e-01` (second column). The probability to proceed to the next state is `5.000000e-01` (third column), other transitions are impossible.

Line 15 End of HMM definition.

The same transition matrix as can be seen in the example is used in all the models. The transition probabilities to stay or proceed to next state are always the same 50% for all the emitting states. (Emitting states are 2–4)

2.6.4 Decoding practically

The SVite general-purpose Viterbi decoder is used to decode phoneme class probabilities. This tool is from STK toolkit and it can be used with the following command:

```
SVite --sourcedict=dict -H svite.hmm -S svite.scp \ -P HTK -l ''*''
-i output.mlf -w gram_net -T 1 -p 5.0 -s 1.0 -t 0.0
```

The meaning of the command parameters is following:

`--sourcedict=dict`	dictionary file, wordlist with the phoneme transcription
`-H svite.hmm`	MMF macro file with HMMs, maps phoneme classifier outputs to HMM phoneme states
`-S svite.scp`	script file, list of the input files
`-P HTK`	target transcription format, use legacy HTK format
`-l ''*''`	dir to store transcription files, affects output MLF, no files will be created there
`-i output.mlf`	output transcriptions to MLF, results will be there
`-w gram_net`	recognise from network, this is word-level grammar network
`-T 1`	number of trace flags
`-p 5.0`	inter-word transition penalty, does not affect isolated words recognition
`-s 1.0`	grammar scale factor, does not affect isolated words recognition
`-t 0.0`	pruning, 0.0 means off

Pruning[1] is a process which reduces quantity of candidate paths during decoding. Beam restriction is applied, any path whose log likelihood falls below threshold is deactivated. The likelihood beam starts by the most probable hypothesis and constant beam width is set by the pruning parameter. The decoding process is also time consuming, almost the same like SFeaCat.

2.7 Recognition evaluation

The final product MLF file contains results for each word, however we need the overall performance results. A HResults HTK tool can be used to obtain them.

```
HResults -z silen -A -I reference.mlf wordlist output.mlf
```

The HTK tool HResults generates nice looking result tables and is capable to output list of mismatched recognitions. The meaning of the parameters is following:

`-z silen`	null class name, used for silence model, it is not a word
`-A`	print command line arguments, common parameter in HTK and STK
`-I reference.mlf`	load Master Label File, this MLF is reference for the comparison
`wordlist`	labelList, list of all the words
`output.mlf`	the SVite product MLF file to compare with the reference

2.8 N-best recognition

The key feature for practical use of recognizer in electronic dictionaries in N-best recognition. On the output is not just the most likely word, but the set of hypotheses ordered by estimated likelihoods.

However the N-best recognition is currently not supported by SVite utility directly. But it is possible to output orientated graph with estimated inter-word transition costs called *lattice* which is based on the word-level grammar network. The graph contains reduced amount of through-paths, the reduction is performed by *pruning*[1].

This orientated graph is stored in the *Standard Lattice Format* (SLF)[1] and afterwards converted to the Master Label File format (MLF). A perl script was made to do this work.

It is important to base the final order on a sum of all the three likelihoods, the word and both surrounding silence model likelihoods. If only the word likelihood is used the short words would get an advantage in ranking.

2.9 Context dependent phonemes

An experimental system with *context dependent phoneme* classifier was built to work with 16KHz speech data. The two neighbour phonemes are considered as the context. The number of possible contexts is too big (theoretically in our case 91125, phoneme count powered by 3).

Each phoneme is modeled by three states, the middle one is the phoneme itself, the side states are phone-to-phone transitions. The amount of the possible transitions is much smaller (1980). Some of these combinations are not existing in natural language, but even without them it is still big number for modeling.

The used triphone estimator evaluates 555 phoneme state classes on output. Which are used in 1264 HMM models, the HMMs are shared by many contextual phonemes, they are *tied* in a group under one HMM. The groups are specified in *tied list* with this format:

```
line = { lc-phn+rc [lc-phn+rc] }
```

lc = left context, rc = right context, phn = phoneme

The first contextual phoneme is the phoneme itself, the second is the reference to the group leader. The system works with totally 89102 context depending phonemes which are tied into 1264 groups.

The decoding of context dependent phonemes needs different HMM definitions. A script which extracts the HMM definitions from GMM model and establishes the correct bindings the was made. This script needs a list of output labels of the MLP as well. The system used precompiled recognition network which was created from grammar network by expansion with dictionary.

The context dependent recognition system generates bigger files with the class probabilities, thus the small reduction of speed is expected. Modelling by context dependent phonemes rises the amount of necessary training data but ensures better accuracy.

Chapter 3

VTLN speaker adaptation

Vocal Track Length Normalisation VTLN[1] is a simple speaker adaptation technique. It is performed on the *feature extraction* level.

3.1 Theory

The speech theory says that the length of vocal tract directly affects the *fundamental tone* frequency which influences all the other speech frequencies. The principle is to shift these frequencies always to the same interval by scaling the frequencies with a single coefficient called *warp factor*. The warp factor lies in the interval $< 0.8, 1.2 >$.

However this factor cannot be estimated analytically. All the waveforms from a single speaker must be processed several times to find the optimal value. The criteria to find the optimal warp factor is to find the maximum of function:

$$\mathrm{Avg}(s) = \frac{\sum_{i=1}^{N} f(w_i, s)}{N} \tag{3.1}$$

Function f returns log likelihood of a certain word w_i for given warp factor s. The set of words has N elements. The average likelihood is directly affected by warp factor.

The *forced alignment* is performed to get these likelihoods. The recognition network is generated by SVite from supplied reference MLF instead of word-level grammar network. To perform forced alignment use the parameter `-I REF_MLF` instead of the `-w NETWORK` parameter.

Forced alignment is much faster than classical recognition, however to get the average likelihood of the set it is necessary to:

1. Choose the warp factor
2. Perform feature extraction with this factor
3. Estimate the phoneme probabilities
4. Perform forced alignment
5. Evaluate the likelihoods

The phoneme estimation phase is a problem, it costs a lot of time. The search of the optimal warp factor for a single speech set of 995 files takes on Athlon 1800+ XP processor 8 hours to few days. The time depends on the waveform lengths and search algorithm.

Another approach to warp factor estimation is to search for it by recognition. The warp factor which produces the maximal word accuracy is taken. This possibility is even slower than the forced alignment approach, but the results are better.

The optimal warp factor is set in the HCopy configuration file by the parameter:

```
HPARM: WARPFREQ = [float number]
```

Since the frequencies are linearly scaled by the warp factor, the effective interval of final spectrum will change compared to the original one. This may confuse the recognition. To prevent this, the lower boundary frequency of the analysis `LOFREQ` and the upper boundary frequency `HIFREQ` are always mapped to themselves. The regions in which the warping function deviates from the linear warping with factor α are controlled with the two configuration variables `WARPLCUTOFF` f_L and `WARPUCUTOFF` f_U. These variables are set in the HCopy config file. On the figure 3.1 is shown the overall shape of the resulting piece-wise linear warping functions.

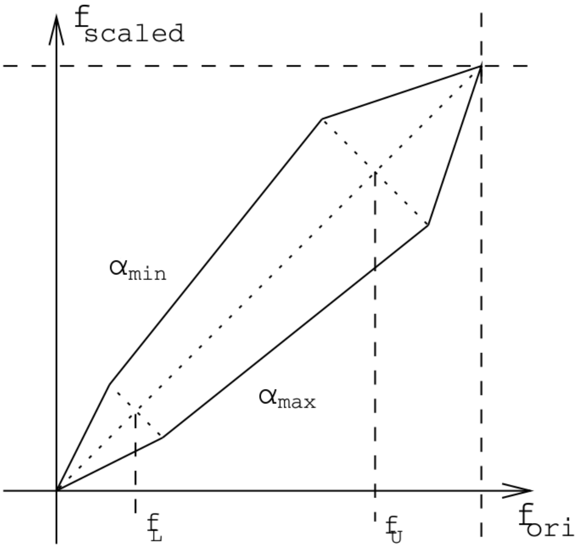


Figure 3.1: Frequency warping

In our case the WARPUCUTOFF and WARPLCUTOFF were both set to 3400Hz.

3.2 Search algorithms

The task is to find a maximum of the function (3.1). First it is necessary to get more information about shape of this function. For this purpose were chosen three speech sets. The first with clean speech, the second with silent speech and the third with strong disturbance.

Function shapes were evaluated with 0.004 warp factor step. Altogether 300 forced alignments were generated to get the graphs of *average word likelihood* on *warp factor* dependency. The shape for clean speech set can be seen in the figure 3.2, for silent speech set it is in the figure 3.3, strange behaviour of this functions is in the figure 3.4.

For good speech data, the gained shape is similar to a round hill without depressions, difference of likelihood extremes is $d \cong 16.5$. For data of poor quality, low hill with small

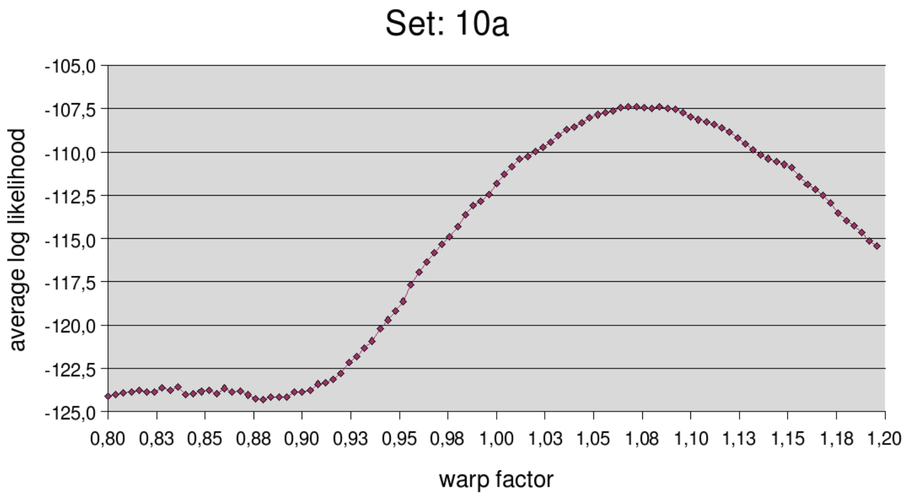


Figure 3.2: Warp function shape, clean speech data

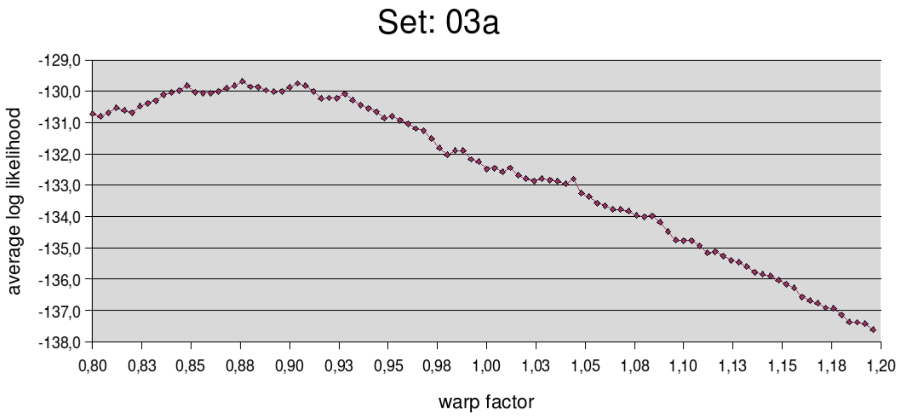


Figure 3.3: Warp function shape, silent speech data

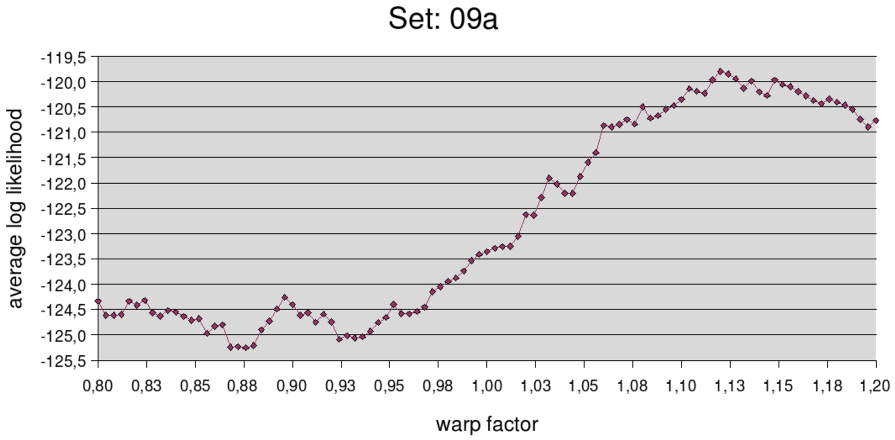


Figure 3.4: Warp function shape, disturbed silent speech data

peaks and small local depressions is returned, the likelihood difference is smaller, $d \cong 8$. In case of the set 09a the shape is roughly disturbed, the maximal difference is only $d \cong 5.5$.

Now we know more about the function, but which search algorithm is optimal? At least the three following approaches can be used or combined.

1. **Equidistant search**

   The whole interval $< 0.8, 1.2 >$ will be tested with constant step, the best result is taken.

2. **Gradient search**

   Two closely situated warp factors near the middle of the search interval are evaluated. The interval is cut in the point of the worse factor. The part with the better factor becomes new search interval.

   The algorithm iterates until the search interval is smaller than given ε value.

   This algorithm needs clean function shape to work well, which is not always true. How small should be the difference between the two warp factors? It should be good to define it as fraction of the current search interval width.

3. **"Left-right around max" search**

   First the warp factor 1.0 is evaluated and in the points 0.8 and 1.2 are set the break-points. The so-far-best result is taken and the point in the middle of the range to its neighbour point is evaluated. The right neighbour is chosen in the even steps, left in odd steps. The algorithm iterates until the search interval (distance between left and right neighbour of the best point) is smaller than given ε value.

   The forced alignment iteration can be spared when two or more better evaluated points in a row are found, thus less time is needed compared to gradient search.

The numbers of forced alignments which are needed to estimate the warp factors with the precision $\varepsilon = 0.004$ are in table 3.1.

method	steps
equidistant search	100
gradient search	18
left-right around max	13 approx.

Table 3.1: Number of steps for VTLN factor estimation with precision $\varepsilon = 0.004$

3.3 Practical usage issues

As mentioned above the application of VTLN is rather time demanding. The second problem is the necessity to have a lot of single speaker data at the beginning of the adaptation.

Another approach to VTLN factor estimation was examined as an attempt to suppress these problems. If it is possible to estimate VTLN factor for each word separately and then use the average value, the VTLN factor could be precised later. However the difference between the average of 995 per-word warp factors and the all-set warp factor was too big. For results see table 3.2

Side product of this experiment is additional knowledge about the sets. The per-word warp factors can be visualised in histograms. Their shape contains information how the warp factors based on small word subset would look like, if we approximate them as an average value. For the histograms see the figure 3.5.

set	classical over-995	per-1-average	difference
04a	0.90	0.96	0.06
06a	0.81	0.97	0.16
07a	0.97	0.98	0.01
09a	1.12	1.01	0.11

Table 3.2: VTLN factor estimation per average

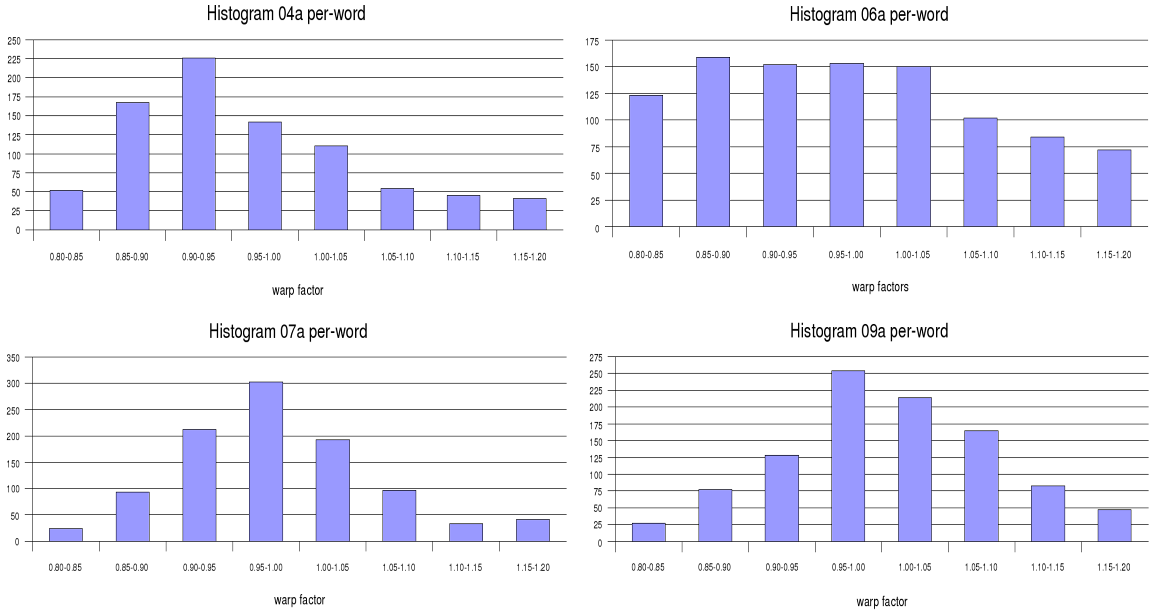


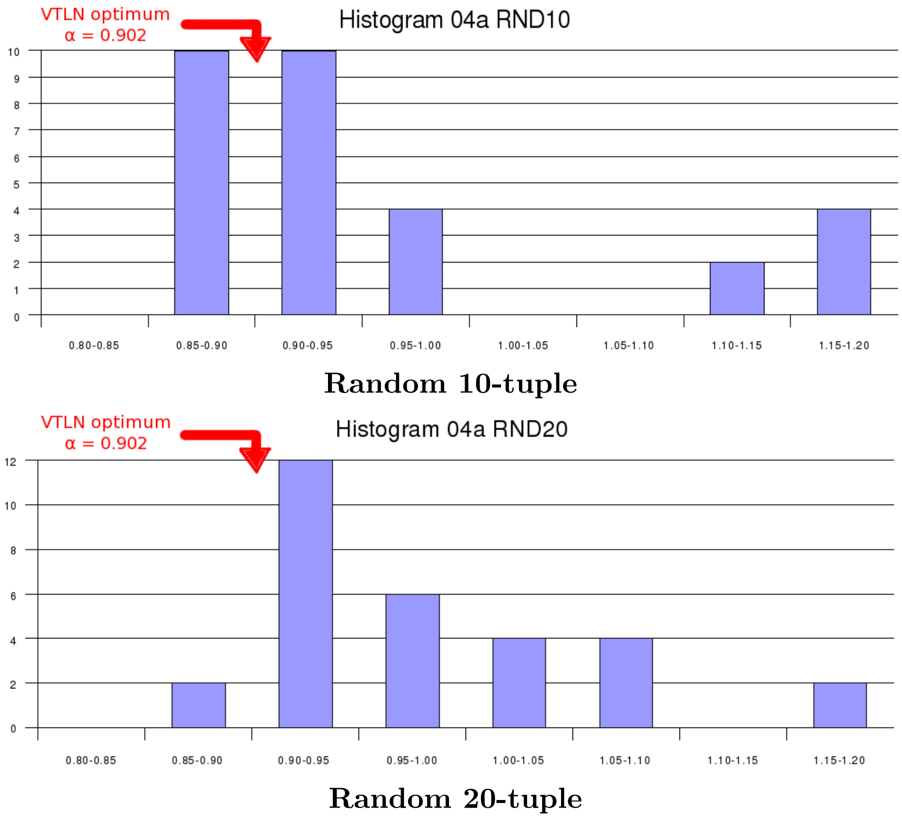
Figure 3.5: Per-word VTLN factor histograms

Practically it is uncomfortable for user to read hundreds of words to estimate the warp factor properly. Thus the focus was put on the examination of the influence of limited data volume to the VTLN factor estimation.

The experiment with random selection of 10-tuples and 20-tuples from the set 04a has been done with 30 trials. For statistical results see the table 3.3, for histograms see the figure 3.6. As supposed, 20 words are not enough to initialize the warp factor properly.

Set:04a	Difference to sets global VTLN factor		
N-tuple	Average	Minimum	Maximum
10	0.07	0.00...	0.29
20	0.08	0.00...	0.28

Table 3.3: VTLN factor estimation from limited speech amount



Random 10-tuple

Random 20-tuple

Figure 3.6: VTLN factor histogram, 30 trials

3.4 VTLN Results

For the VTLN warp factor search method evaluation was used phoneme classifier trained on the VTLN speech data. The used phoneme classifier had hierarchical design with split time context.

The average performance results for each VTLN factor search method are in the table 3.4 in the rightmost column. For complete per-speaker recognition results see the appendix table 5.3, page 36. The results will be commented further in the chapter 5.2.

Method	set 03a		set 09a		set 10a		Avg.
	warp f.	acc.[%]	warp f.	wcc.[%]	warp f.	acc.[%]	acc.[%]
Equidistant search	0.87	1.9	1.11	2.7	1.07	58.6	
Gradient search	0.90	1.8	1.12	2.6	1.06	58.8	52.7
Left-right search	0.87	1.9	1.12	2.6	1.07	58.6	52.7
Recognition search	0.88	1.9	0.96	9.1	1.06	58.8	53.9

Table 3.4: optimal VTLN warp factors

In the table 3.4 can also be seen the estimated warp factors for selected data sets. The factors shouldn't differ within the columns. The speech sets 03a and 09a are poor quality thus the factors differ, the set 10a gains very good recognition results and its warp factors are almost the same.

The VTLN factors were estimated with the precision $\varepsilon = 0.004$. For complete per-speaker table of VTLN factors see the appendix table 5.4, page 36.

The effect in performance of the VTLN incorporation into the system will be described further in the chapter 5.

Chapter 4

Experiments

Experiments with 5 different system setups were performed. The performance evaluation is done from all the speech sets with the exception of the set 03a due to poor record quality.

1. For the first experiment the test data were downsampled to 8KHz sampling frequency. Parametrized to 15 critical bands FBANK features and mean-variance normalized. The phonemes were classified by 4-Layer MLP based classifier with 135 phoneme state classes.

2. For the second experiment were used 16KHz test data. Parametrized to 23 critical bands FBANK features. The phonemes were classified by split time context classifier with three 3-Layer MLPs inside. Each phoneme was modelled by 3 states, totally 135 outputs.

3. The third experiment setup is similar to the second. The change is in the usage of the VTLN normalized features.

4. The fourth experiment setup is also similar to the second. The change is in the usage of the both VTLN and mean-variance normalized features as input of the composite classifier.

5. For the fifth experimental setup were used 16KHz test data parametrized to FBANK with 23 critical bands and VTLN normalisation. The features were mean-variance normalized and classified by a context dependent phoneme classifier based on a single 4-Layer MLP. The classifier distinguishes 555 classes based on context dependent phoneme states. The decoder used a grammar network with triphone expansion of words.

For the recognition results of each setup see the table 4.1 and the graph in the figure 4.1. For detail per-speaker results of the fifth setup, which performed the best results, see the figure 4.2. The speaker 03a is not included in average results from above.

				1-BEST	5-BEST	10-BEST	20-BEST
fvz	VTLN	norm.	note	WAcc avg	WAcc avg	WAcc avg	WAcc avg
8000	NO	YES	model trained with VTLN data	47,85%	64,48%	70,50%	75,90%
16000	NO	NO	---	52,54%	67,79%	73,27%	78,31%
16000	YES	NO	---	52,73%	66,66%	71,69%	76,18%
16000	YES	YES	---	52,15%	66,35%	71,68%	76,79%
16000	YES	YES	triphones, ANN555 outputs	58,48%	72,56%	78,71%	83,69%

Table 4.1: Comparison of the different system setups

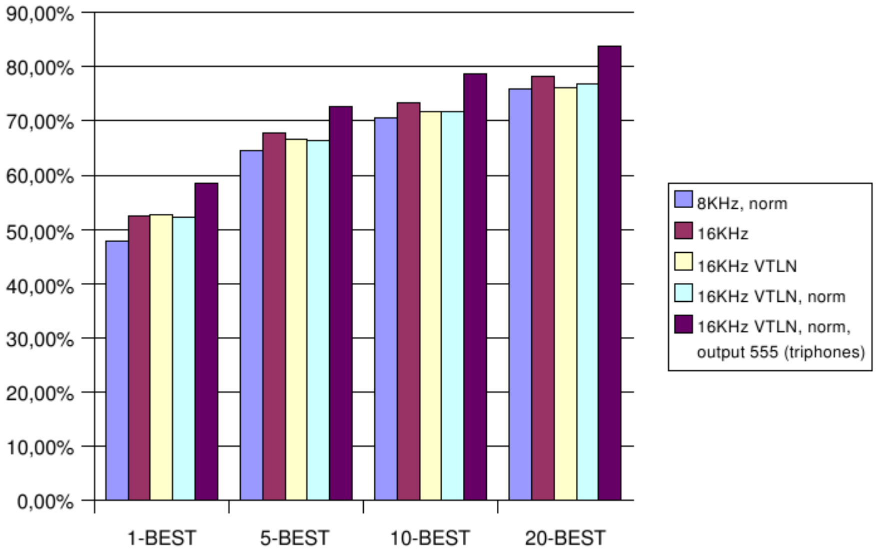


Figure 4.1: Comparison of the different system setups

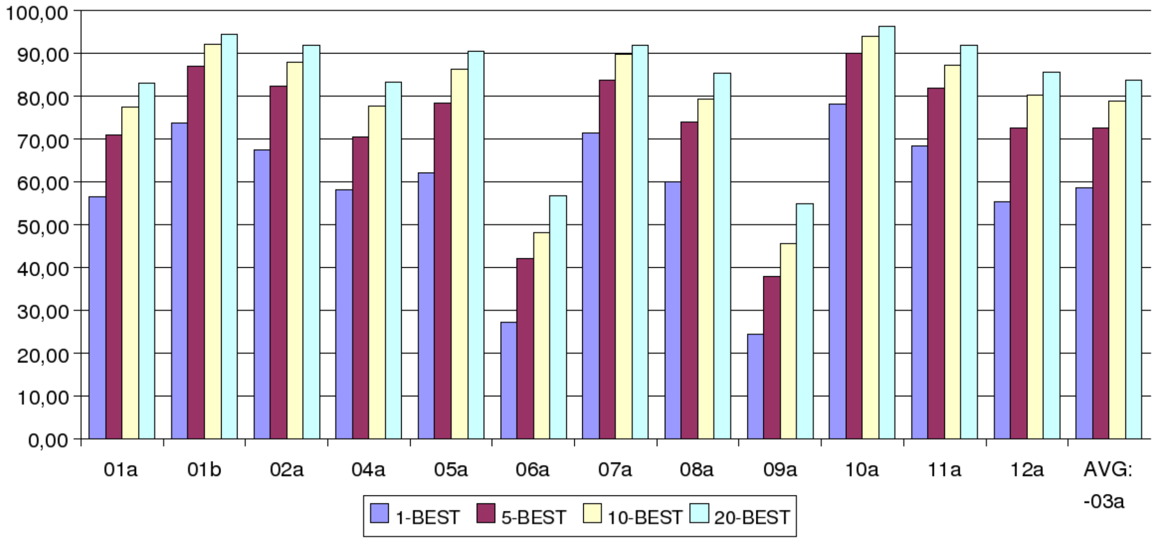


Figure 4.2: N-BEST per-speaker results, best setup (context dependent phn.)

Chapter 5

Conclusion and discussion

The performed experiments has shown the very positive effect of the N-best recognition to word accuracy which can be seen in the figure 4.2. With higher N-best level, the word accuracy always improves. The biggest accuracy growth is between the 1-best to 5-best recognition. As as optimal N-best level can be taken 10-best, because it is still suitable for users to search from 10 options and the further accuracy growth is not as big as for low N-s. The average percentual accuracy improvement from the 1-best level to 10-best level is 20.24%.

The word accuracy improvement is almost the same both for the poor quality and the high quality record sets. So that even for the poor quality sets 06a and 09a, whose 1-best accuracies are around 25%, the accuracy on 20-best level gets over 50%, which means that on this level more than every second word will be correctly recognized. The N-best recognition helps the system to be more robust to the poor quality data.

If we look at the graph on the figure 4.1, we see that the system which analyzed 8KHz test data gained almost as good performance as the 16KHz systems. The difference is only 2.7% on 10-best recognition. The 8KHz test data were simulating the telephone channel influence. The other remarkable fact is that the overall performance has dropped by 1.6% (10-best) with the VTLN incorporation. The general improvement within most of the speech sets was pulled down by the three sets whose accuracies had dramatically dropped, see the appendix figure 5.2 for details. With the incorporation of the mean-variance normalization, the average recognition accuracy remained almost on the same. Some sets improved the results, some decreased the performance, for details see the appendix figure 5.2. The very good performance improvement 7.0% (10-best) was gained by the system based on the context dependent phoneme modeling, compared to the 16KHz phoneme state system with VTLN and feature normalisation.

The overall performance 78.71% on the 10-best level is sufficent to use the recognizer in electronic dictionaries for word look-up. To give better comfort to the user, it would be good to implement on demand speech synthesis of search results. The user will be able to correct his pronunciation or distinguish the one word he had in mind.

The performance can even improve with the use of dictionary with more pronunciation variants. On the other hand the performance would probably decrease by usage of the large vocabulary because of larger confusion among the words. Both modifications are needed for use in real application, but both were irrelevant for the different setup comparison which was the main purpose of this project.

5.1 Compare GMM x Hybrid

A parallel project based on the same test data was done. It was based on the classical GMM–HMM approach with VTLN, mean-variance feature normalisation and HLDA optimisation. The same test speech database allows us to compare the hybrid to the GMM–HMM system performance.

The general advantages of *hybrid recognition* are:

- less computational complexity than via conventional GMM recognition
- better hardware support to matrix multiplications in MLPs than to Gaussian mixtures
- lower requirements on statistical characters on input (GMM requires statistically independent features)
- input can be mixture of feature types
- simple incorporation of acoustic context
- smaller system with less features

For a number of tasks, simple tied-Gaussian mixture systems do not perform as well as hybrid ANN–HMM systems that use a similar number of parameters and the same input features. For equivalent performance, the classical HMM system must be made much more complicated (for instance, typically using context-dependent models and many more parameters).

Current state-of-the-art GMM-HMM systems outperform the best hybrid systems on many tasks, but the HMM systems are frequently more complicated[2].

In the figure 5.1 can be seen the recognition performance comparison of the best hybrid system to the 16KHz GMM–HMM system from the parallel project. The GMM–HMM system includes both VTLN and feature normalisation and uses context dependent phonemes.

The complexity of models can be expressed by the number of trained features. In hybrid case, the model contains 999 951 features. In GMM–HMM case, the model contains 6 127 636 features. The more complex model is a good premise for better recognition accuracy.

The results were compared on the 10-best recognition level. The GMM–HMM system performed with better accuracy in case of 10 sets, the most distinctive difference is in the case of set 09a. For 2 of the sets the hybrid system gained better results. However the average result of GMM–HMM system is better by 8.91%.

5.2 VTLN discussion

Now let's get back to the VTLN theme, the following lines are explaining the appendix table 5.3 on the page 36.

On the line 1 there are the results for the reference VTLN factor $\alpha = 1.0$. All the following results shouldn't be lower than values on this line. The following two lines contains the word accuracy gained with warp factors estimated by gradient search (line 2) and "left-right" search (line 3). The fourth line contains results of the search based on recognition instead of forced alignment, gradient search approach was used.

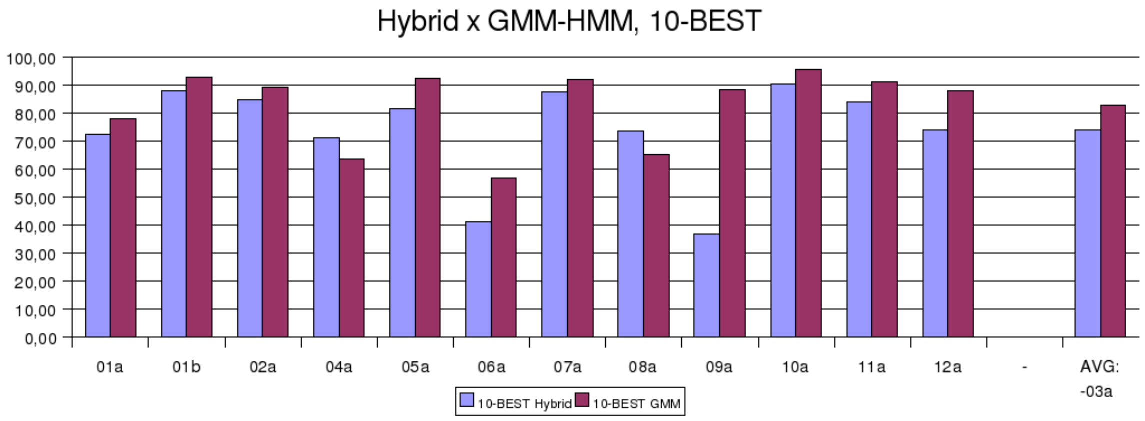


Figure 5.1: Hybrid x GMM–HMM system comparison

The final summary is, that in average the gradient search and the "left-right" search gain equivalent results. The important fact is that the "left-right" search needs less iterations to gain the same VTLN factor precision. The recognition based search gains even better results, which were due to performance improvement of 6 speech sets, but only two sets had improved by more than 2%. The recognition based search needs approximately twice as more time compared to forced alignment based one. This makes the "left-right" search algorithm the best one for practical use.

In the table 5.3 (page 36) can be seen a strange result in the case of 09a and 12a sets. Curious is that the word accuracy has dropped under the reference value. In the 12a the drop was tiny, perhaps the reason is the VTLN factor precision error. In the case of the set 09a the drop was more dramatic, the reason will be explained below.

In the appendix table 5.4 (page 36) are all the estimated VTLN factors. The differences within the columns should be very small. The most interesting is to compare the forced alignment based searches (lines 1-3) to the recognition based search (line 4). The difference is remarkable for the sets 09a and 06a. The performance of these sets is quite poor thus the VTLN factor estimation has problems with likelihood to performance correlation. In the case 09a there is a major difference $d = 0.16$ between the warp factors.

That's why the focus was put on the set 09a to find out more and explain these strange results. Normally the *word accuracy* is correlated with *average word likelihood* as shown on the page 37 in the figure 5.5. This is not true in case of 09a, for all the graphs see the figure 5.6. The reason should be searched in the input waveforms. The listening inspection has discovered the possible reason. The waves are disturbed by wide band noise in the middle frequencies similar to PC-fan, and the speech is quite slow and quiet, this makes the silence model too strong and confuses the forced alignment and recognition.

The speech set 09a was discussed with my colleague who works on the parallel GMM-HMM project, and there the set 09a gains one of the best results, that is really very curious. The possible reason is that his GMM-HMM system is more robust to noisy acoustic environment. It also uses different features (PLP) which are based on spectrum scope, whereas our features look first over the time in one frequency band and then only encoded spectral information is preserved to MLP.

At the end of the VTLN discussion it should be repeated that the usage of VTLN in real application hybrid recognizer is disputable. The biggest problems are the speech amount demand and the time complexity to get the proper VTLN factor.

Also the influence of mismatched pronunciation from English beginner should contribute to the wrong VTLN factor estimation. The possible solution is to record a longer sentence in speakers native language. This would require to store models for other languages which should be uncomfortable.

5.3 Final resume

At this very final part of this document I would like to state, that all the main goals from the assignment were successfully reached. The big part of this success belongs to my supervisor, who had never hesitated to give me as much support as possible, who explained me parts of the theory and who gave me good advices how to go ahead in the work.

The main target of this project was to build hybrid isolated word recognition system and to evaluate the performance of several systems with MLP based phoneme class models. Some of these models used the VTLN speaker adaptation feature normalisation or the contextual phoneme modeling.

All the system setups worked well and the results fulfilled the good expectations. The most important is the very good influence of N-best on recognition accuracy, which allows to use the current system in electronic dictionaries.

In this document it has been tried to introduce the hybrid speech recognition from the practical side. I've been told that the similar projects will be done in the future. I hope this document would be good initial point for the speech interested students and will provide them useful information for their future projects.

Bibliography

[1] Young, S., Jansen, J., Odell, J., Ollason, D., Woodland, P.: The HTK book, Entropics Cambridge Research Lab., Cambridge, UK, 2002

[2] Morgan, N., Bourlard, H.:An Introduction to Hybrid HMM/Connectionist Continuous Speech Recognition, IEEE Signal Processing Magazine, pp. 25-42, May 1995.

[3] Schwarz, P., Matejka, P., Cernocky, J.: Hierarchical structures of neural networks for phoneme recognition, In: Proceedings of ICASSP 2006, Toulouse, FR, 2006

[4] Schwarz, P., Matejka, P., Cernocky, J.: Towards Lower Error Rates in Phoneme Recognition, In: Proceedings of 7th International Conference Text,Speech and Dialoque 2004, Brno, CZ, Springer, 2004

[5] Allen, J. B.: How do humans process and recognize speech?, IEEE Trans. on Speech and Audio Processing, 2(4):567-577, 1994

Abbreviation list

ANN - artificial neural network

ASR - automatic speech recognition

CRBE - critical band energies

DCT - discrete cosine transform

FBANK - filter bank

FFT - fast Fourier transform

GMM - Gaussian mixture model

HMM - hidden Markov model

HTK - hidden Markov toolkit

MLF - master label file

MLP - multi-layer perceptron

MMF - master model file

PER - phoneme error rate

SLF - standard lattice format

TRAPs - TempoRAl Patterns

VTLN - vocal tract length normalisation

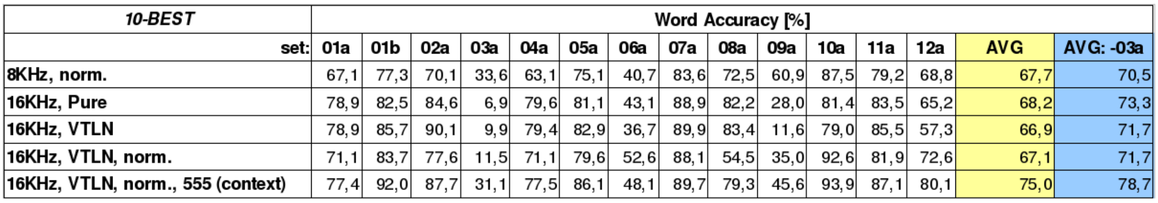
WER - word error rate

Appendix list

1. System setup per-speaker comparison
2. Results compared to subjective data quality
3. VTLN per-speaker results
4. CD appendix

Appendix

System setup per-speaker comparison



10-BEST	Word Accuracy [%]														
set:	01a	01b	02a	03a	04a	05a	06a	07a	08a	09a	10a	11a	12a	AVG	AVG: -03a
8KHz, norm.	67,1	77,3	70,1	33,6	63,1	75,1	40,7	83,6	72,5	60,9	87,5	79,2	68,8	67,7	70,5
16KHz, Pure	78,9	82,5	84,6	6,9	79,6	81,1	43,1	88,9	82,2	28,0	81,4	83,5	65,2	68,2	73,3
16KHz, VTLN	78,9	85,7	90,1	9,9	79,4	82,9	36,7	89,9	83,4	11,6	79,0	85,5	57,3	66,9	71,7
16KHz, VTLN, norm.	71,1	83,7	77,6	11,5	71,1	79,6	52,6	88,1	54,5	35,0	92,6	81,9	72,6	67,1	71,7
16KHz, VTLN, norm., 555 (context)	77,4	92,0	87,7	31,1	77,5	86,1	48,1	89,7	79,3	45,6	93,9	87,1	80,1	75,0	78,7

Table 5.1: System setup comparison, per-speaker

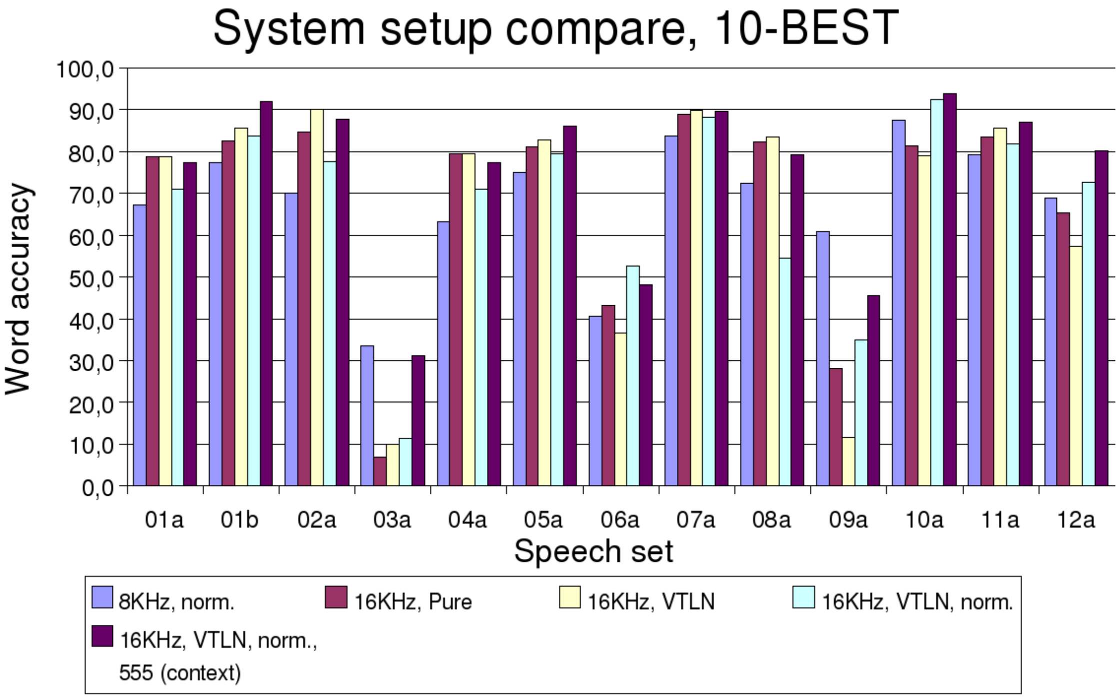


Figure 5.2: System setup comparison, per-speaker

Results compared to subjective data quality

Set	Quality description	Sex
01a	overexcited microphone, power-line hum, microphone blowing, background noises	female
01b	power-line hum, background noises	female
02a	power-line hum, few words cut incorrectly	female
03a	power-line hum, weak signal, no background noises	female
04a	hum, relatively good SNR	female
05a	good SNR, some hiss	male
06a	overexcited microphone, few records proper	female
07a	perfect SNR, good set	male
08a	weak hum, weak bg. noises	female
09a	medium noise (middles, like PC fan), quiet speech	male
10a	medium noise (middles, like PC fan)	male
11a	power-line hum, background noises	female
12a	strong noise (low-middles, like loud PC case fan)	male

Table 5.2: Speech sets characteristics

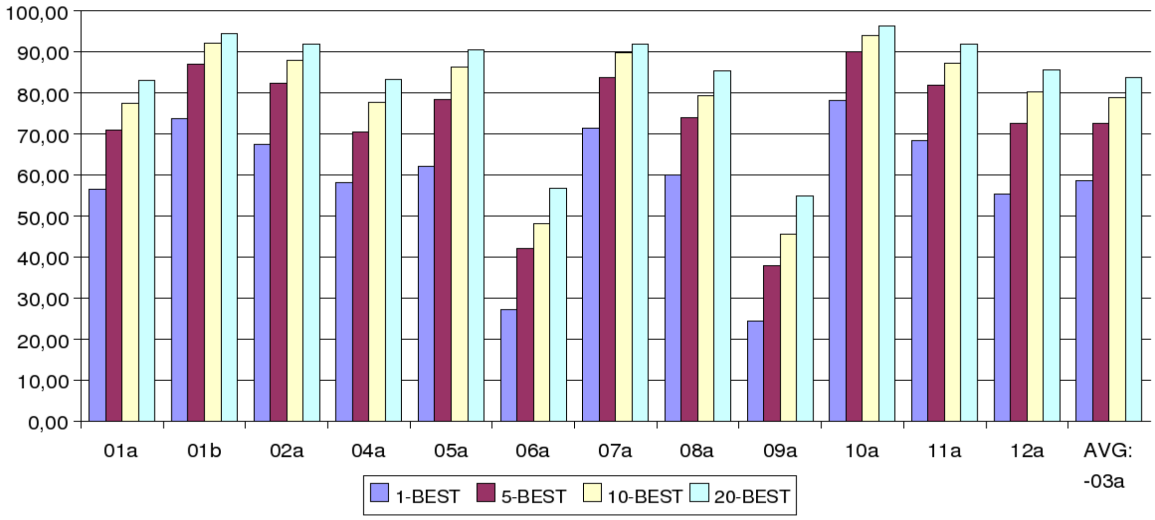
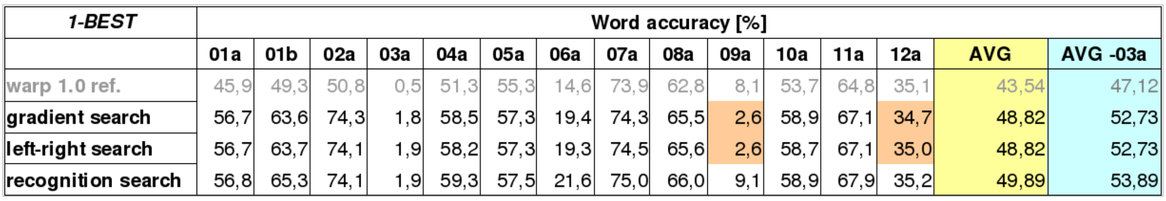


Figure 5.3: N-best per-speaker results, best system setup (context dependent phonemes)

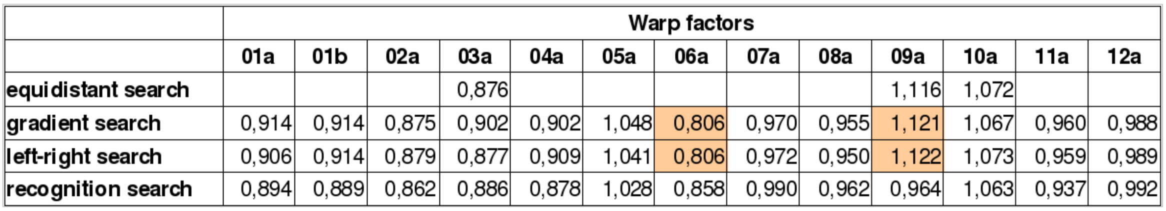
Generally the test data with quiet speech and wide-band disturbance are poorly recognized. Which is the case of the sets 03a and 09a. In case of 06a some records were made with overexcited microphone some with proper setup, this disproportion could cause the degradation of the performance.

VTLN per-speaker results



1-BEST	Word accuracy [%]														
	01a	01b	02a	03a	04a	05a	06a	07a	08a	09a	10a	11a	12a	AVG	AVG -03a
warp 1.0 ref.	45,9	49,3	50,8	0,5	51,3	55,3	14,6	73,9	62,8	8,1	53,7	64,8	35,1	43,54	47,12
gradient search	56,7	63,6	74,3	1,8	58,5	57,3	19,4	74,3	65,5	2,6	58,9	67,1	34,7	48,82	52,73
left-right search	56,7	63,7	74,1	1,9	58,2	57,3	19,3	74,5	65,6	2,6	58,7	67,1	35,0	48,82	52,73
recognition search	56,8	65,3	74,1	1,9	59,3	57,5	21,6	75,0	66,0	9,1	58,9	67,9	35,2	49,89	53,89

Table 5.3: VTLN results



	Warp factors												
	01a	01b	02a	03a	04a	05a	06a	07a	08a	09a	10a	11a	12a
equidistant search				0,876						1,116	1,072		
gradient search	0,914	0,914	0,875	0,902	0,902	1,048	0,806	0,970	0,955	1,121	1,067	0,960	0,988
left-right search	0,906	0,914	0,879	0,877	0,909	1,041	0,806	0,972	0,950	1,122	1,073	0,959	0,989
recognition search	0,894	0,889	0,862	0,886	0,878	1,028	0,858	0,990	0,962	0,964	1,063	0,937	0,992

Table 5.4: estimated VTLN warp factors

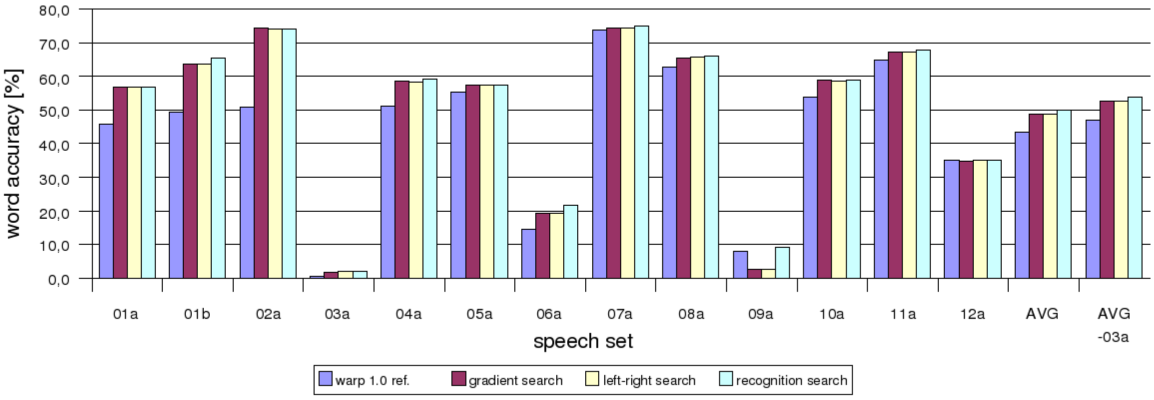


Figure 5.4: VTLN methods - Word accuracy comparison

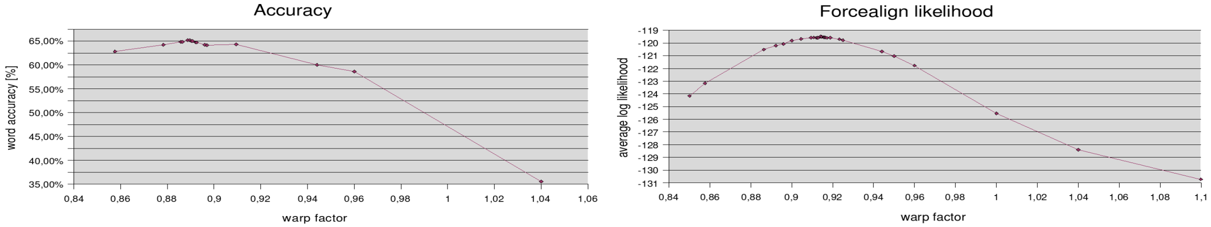
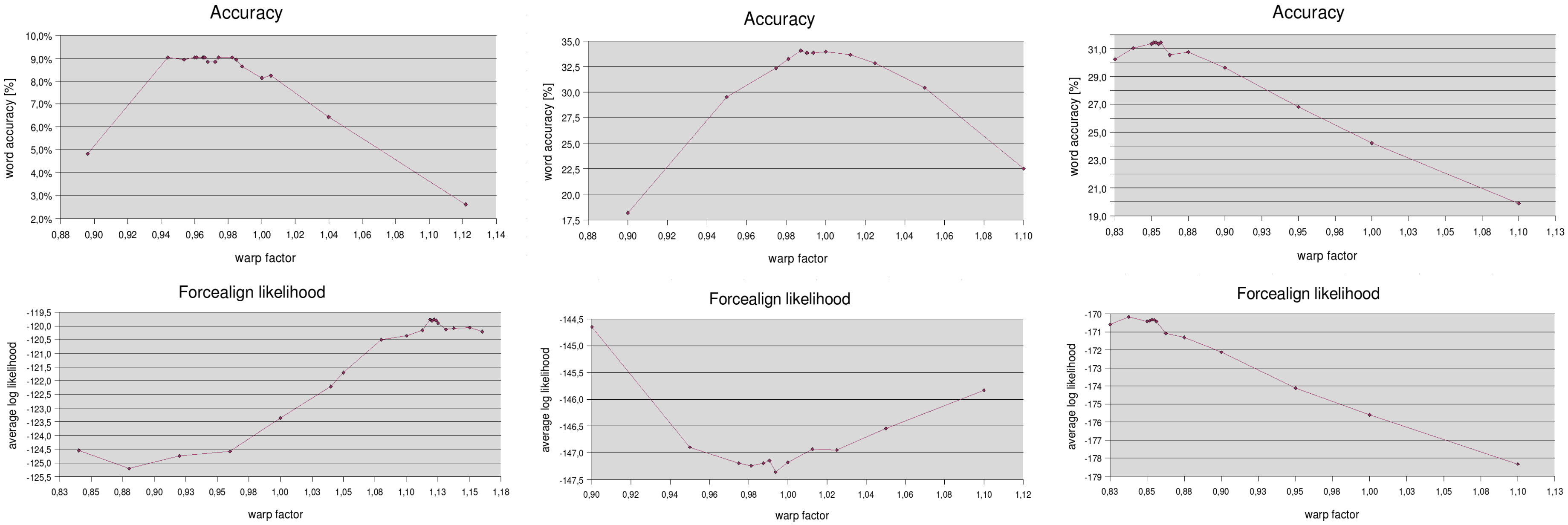


Figure 5.5: VTLN: Normal correlation of word accuracy on average log likelihood (set 01b)



09a - incorrect correlation

09a fea. normalised - incorrect correlation

06a fea. normalised - normal correlation

Figure 5.6: VTLN: The correlation between word accuracy and log likelihood

CD appendix

The appended CD contains:

- HTK toolkit
- STK toolkit
- One 16KHz speech set with features, phoneme class probabilities, MLF results and recognition script as demonstration of hybrid recognition
- Sources of this document
- Published materials for EEICT
- Matlab HTK file visualisation scripts
- All the used scripts and complete results