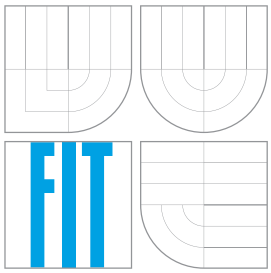
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁVAČ IZOLOVANÝCH SLOV PRO OVLÁDÁNÍ ELEKTRONICKÝCH SLOVNÍKŮ

RECOGNITION OF ISOLATED WORDS FOR ELECTRONIC DICTIONARIES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PAVEL HRDLIČKA
AUTHOR

VEDOUCÍ PRÁCE — Ing. FRANTIŠEK GRÉZL
SUPERVISOR

BRNO 2007

Abstrakt

Tato bakalářská práce se zabývá sestavením rozpoznávače izolovaných slov pro elektronické slovníky. Fonémový rozpoznavač je realizován pomocí HTK (Hidden Markov Model Toolkit). Na začátku tohoto dokumentu jsou stanoveny základní cíle práce. V následující kapitole je teoretický rozbor, který se věnuje procesu rozpoznávání izolovaných slov pomocí skrytých Markovových modelů. Další kapitola se věnuje specifikaci řečových dat, která byla použita pro testování rozpoznávače. Dále jsou zde popsány další prostředky, které byly k dispozici pro sestavení rozpoznavače, jako modely, slovník a gramatika. Před sestavením rozpoznávače bylo třeba vyřešit převod mezi sadou fonémů která byla použita ve slovníku a mezi sadou, kterou používá rozpoznávač. Rozpoznavač byl nejprve sestaven s použitím 8 kHz modelů, později 16 kHz. Byly použity normalizační techniky a technika adaptace na mluvčího. Získaná data byla zpracována a výsledky jsou zhodnoceny v samostatné kapitole. V závěru je diskutováno, zda bylo dosaženo vytýčených cílů a jaké jsou další plány vývoje aplikace.

Klíčová slova

Rozpoznávání slov, HTK, VTLN, PLP, slovník, foném

Abstract

This work is concerned with creation of isolated word recognizer for electronic dictionaires, testing its functionality on data sample and improvement by normalisation and speaker adaptation techniques. Word recognizer is built on HTK (Hidden Markov Model Toolkit). At the beginning of this document, the main aims of the work are set. In the next chapter is theoretical analysis, which describes process of recognition of isolated words with hidden Markov models. Next chapter specifies the speech data, which were used for testing. Other resources for building recognizer, like models, dictionary and grammar are described in next chapter. Before creation of recognizer, it was necessary to solve conversion between the phonemes set which was used in dictionary and set, which uses the recognizer. The recognizer was built with 8 kHz models first, than 16 kHz models were also used. Normalisation and speaker adaptation techniques were used. Obtained data were processed and results are analyzed in separate chapter. Finally is discussed, if the goals of the work were reached and what are the next steps of application development.

Keywords

Speech recognition, HTK, VTLN, PLP, dictionary, phoneme

Citace

Pavel Hrdlička: Recognition of Isolated Words for Electronic Dictionaries, bakalářská práce, Brno, FIT VUT v Brně, 2007

Recognition of Isolated Words for Electronic Dictionaries

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Františka Grézla. Další informace mi poskytl Doc. Dr. Ing. Jan Černocký. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Hrdlička
15th May 2007

Poděkování

Hardware použitý pro tuto práci částečně poskytl CESNET pod projekty č. 119/2004, č. 162/2005 a č. 201/2006. Tato práce byla částečně podporována Ministerstvem průmyslu a obchodu České republiky pod projektem č. FT-TA3/006, Grantovou agenturou České republiky pod projektem č. 102/05/0278 a Ministerstvem školství, mládeže a tělovýchovy České republiky pod projektem č. MSM0021630528.

Contents

Chapter 1

Introduction

Nowadays, the study of foreign languages is very important. This trend must unavoidably show itself in modern information technologies. Electronic dictionaries are able to search big amount of entries in a short time.

The aim of this work is to eke out the classical dictionary control by keyboard and mouse with control by human speech. This approach brings bigger comfort with searching of isolated words. The user enters the searched word and the programme returns couple of possibilities, from which he can choose. This can be also used in case, when the user don´t know, how the word is written. He can choose the best variant and continue work with the dictionary.

The main aim of this work is to adapt user data, which were provided by Lingea company, to existing models. Because only few data were provided, it is not possible to train new models. Instead, provided data are adapted to models already trained on big amount of speech data.

This bachelor work is divided into six chapters. In the next chapter, there are basics of automatic speech recognition. The third chapter specifies dictionary, speech data and models, which were used in experiments. Next chapter describes building of recognizer in detail. First, creation of dictionary is analysed. Than downsampling of test data is solved, where the best downsampling algorithm is discussed. After that follows extraction of Perceptual Linear Prediction (PLP) features and creation of simple grammar. The fifth chapter shows results of experiments with 8 kHz, 16 kHz and 16 kHz + vocal tract length normalisation models in tables and diagrams. Last chapter resumes work done, and implies possibilities of next work on the recognizer.

Chapter 2

Basics of automatic speech recognition

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:

$$\overline{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}, \Theta) \tag{2.1}$$

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. 2.1 can be replaced by a all possible sequences of phones $\mathbf{Q}$ which together form word $\mathbf{W}$:

$$\overline{\mathbf{W}} = \arg\max_{\mathbf{W}} \sum_{Q} P(\mathbf{W}, \mathbf{Q}|\mathbf{X}, \Theta) \tag{2.2}$$

By using Bayes rule we can further obtain:

$$\overline{\mathbf{W}} = \arg\max_{\mathbf{W}} \sum_{Q} \frac{P(\mathbf{X}|\mathbf{W}, \mathbf{Q}, \Theta) P(\mathbf{W}, \mathbf{Q}|\Theta)}{P(\mathbf{X}|\Theta)} \tag{2.3}$$

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:

$$\overline{\mathbf{W}} = \arg\max_{\mathbf{W}} \sum_{Q} P(\mathbf{X}|\mathbf{W}, \mathbf{Q}, \Theta) P(\mathbf{Q}|\mathbf{W}, \Theta) P(\mathbf{W}|\Theta) \tag{2.4}$$

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

[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 around forty-five of them for English.

Θ into parts each of which will affect the probability term it is contained in:

$$\overline{\mathbf{W}} = \arg\max_{\mathbf{W}} \sum_{Q} P(\mathbf{X}|\mathbf{Q}, \Theta_{AM}) P(\mathbf{Q}|\mathbf{W}, \Theta_{PM}) P(\mathbf{W}|\Theta_{LM}) \tag{2.5}$$

The three probability models in Eq. 2.5 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.

2.1 Parametrisation

In Perceptual Linear Prediction (PLP) technique [4], several well known properties of hearing are simulated by practical engineering approximations and the resulting auditory-like spectrum of speech is approximated by the autoregressive all-pole model. A block diagram of the PLP method is shown in 2.1.

- **Spectral analysis**

  The speech segment is weighted by the Hamming window

  $$W(n) = 0.54 + 0.46\, cos(2\pi n/(N-1)), \tag{2.1}$$

  where N is the length of the window.

  The typical length of the window is about 20 ms. The Discrete Fourier Transform DFT transforms the windowed speech segment into the frequency domain. Typically, the Fast Fourier Transform FFT is used here. For a 10 kHz sampling frequency, a 256-point FFT is needed for transforming the 200 speech samples from the 20-ms window padded by 56 zero-valued samples. The real and imaginary components of the short-term speech spectrum are squared and added to get the short-term power spectrum.

  $$P(\omega) = Re(S(\omega))^2 + Im(S(\omega))^2. \tag{2.2}$$

- **Critical-band spectral resolution**

  The spectrum $P(\omega)$ is warped along its frequency axis ω into the Bark frequency Ω and integrated into filters.

- **Equal-loudness preemphasis**

  The sampled $\Theta(\Omega(\omega))$ is preemphasized by the simulated equal loudness curve.

- **Intensity-loudness power-law**

  The last operation prior to the all-pole modeling is the cubic-root amplitude compression

  $$\Phi(\Omega) = \Xi(\Omega)^{0}.33 \tag{2.3}$$

  This operation is an approximation to the power law of hearing and simulates the non-linear relation between the intensity of sound and its perceived loudness. Together with the psychophysical equal-loudness preemphasis, this operation also reduces the spectral amplitude variation of the critical-band spectrum so that the following all-pole modeling can be done with relatively low model order.

- **Autoregressive modeling**

  In the final operation of the PLP analysis, $\Phi(\Omega)$ is approximated by the spectrum of an all-pole model using the autocorrelation method of an all-pole spectral modeling. Details of the spectral all-pole modeling are sufficiently well described elsewhere and we give here only a brief overview of its principle: The inverse DFT IDFT is applied to $\Phi(\Omega)$ to yield the autocorrelation function dual to $\Phi(\Omega)$. Typically, a 34 point IDFT is used. The IDFT is the better choice here than the inverse FFT, since only a few autocorrelation values are needed. The rst M+1 autocorrelation values are used to solve the Yule-Walker equations for the autoregressive coeffitcents of the M-th order all-pole model. The autoregressive coefficients could be further transformed into some other set of parameters of interest, as is the set of cepstral coeffcients of the all-pole model.

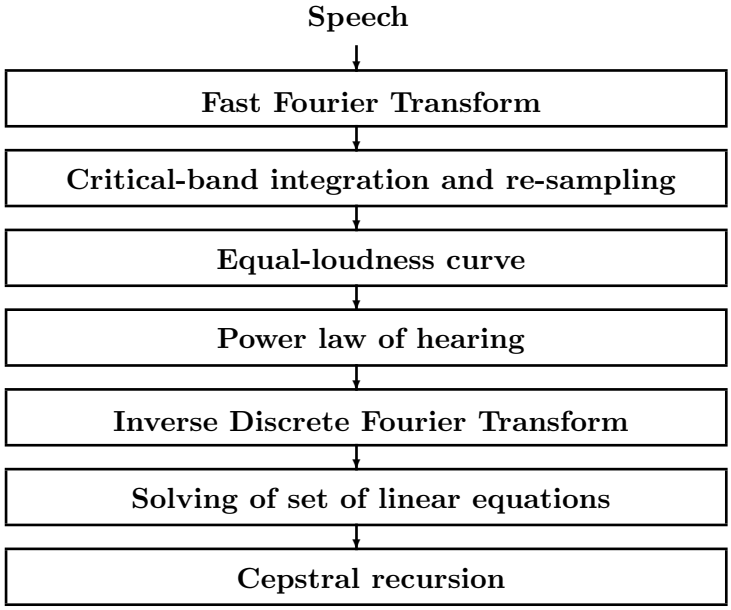


Figure 2.1: Steps in the computation of PLP [3]

2.2 Hidden Markov models

In HMM based speech recognition, it is assumed that the sequence of observed speech vectors corresponding to each word is generated by a Markov model as shown in Fig. 2.2. A Markov model is a finite state machine which changes state once every time unit and each time t that a state j is entered, a speech vector o_t is generated from the probability density $b_j(o_t)$. Furthermore, the transition from state i to state j is also probabilistic and is governed by the discrete probability a_{ij} . Fig. 2.2 shows an example of this process where the six state model moves through the state sequence $X = 1, 2, 2, 3, 4, 4, 5, 6$ in order to generate the sequence o_1 to o_6. Notice that in HTK, the entry and exit states of a HMM are non-emitting. This is to facilitate the construction of composite models as explained in more detail later. The joint probability that O is generated by the model M moving through the state sequence X is calculated simply as the product of the transition probabilities and the output probabilities. So for the state sequence X in Fig. 2.2

$$P(O, X|M) = a_{12}b_2(o_1)a_{22}b_2(o_2)a_{23}b_3(o_3)... \tag{2.1}$$

However, in practice, only the observation sequence O is known and the underlying state sequence X is hidden. This is why it is called a Hidden Markov Model.

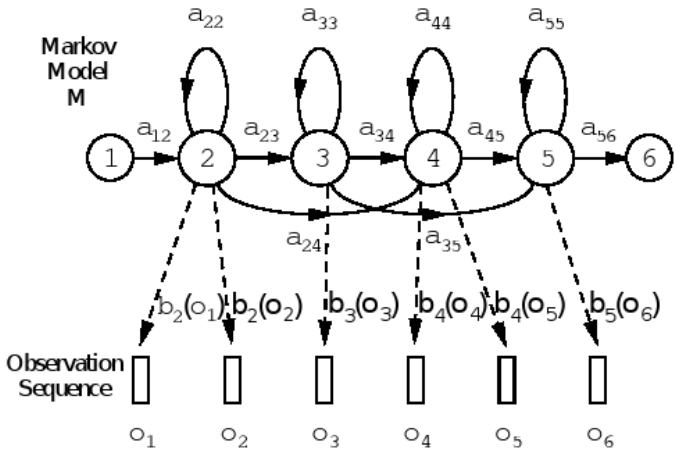


Figure 2.2: The Markov Generation Model

Given that X is unknown, the required likelihood is computed by summing over all possible state sequences $X = x(1), x(2), x(3), ..., x(T)$ that is

$$P(O, X|M) = \sum_X a_{x(0)x(1)} \prod_{T=1}^{T} b_x(t)(o_t)a_{x(t)x(t+1)} \tag{2.2}$$

where $x(0)$ is constrained to be the model entry state and $x(T+1)$ is constrained to be the model exit state.

2.3 Vocal tract length normalisation

Vocal tract length normalisation (VTLN) [6] aims to compensate for the fact that speakers have vocal tracts of different sizes. VTLN can be implemented by warping the frequency axis in the filterbank analysis. In HTK simple linear frequency warping is supported. The warping factor α is controlled by the configuration variable WARPFREQ. Here values of $a < 1.0$ correspond to a compression of the frequency axis. As the warping would lead to some filters being placed outside the analysis frequency range, the simple linear warping function is modified at the upper and lower boundaries. The result is that 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 a are controlled with the two configuration variables (WARPLCUTOFF – f_L) and (WARPUCUTOFF – f_U). Figure 2.3 shows the overall shape of the resulting piece-wise linear warping functions.

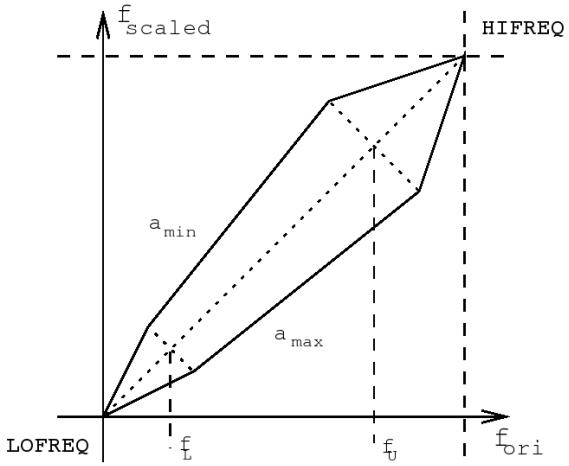


Figure 2.3: Frequency Warping

The warping factor α can for example be found using a search procedure that compares likelihoods at different warping factors. A typical procedure would involve recognising factors in the range 0.8 - 1.2. The factor that gives the highest likelihood is selected as the final warping factor. Instead of estimating a separate warping factor for each utterance, large units can be used by for example estimating only one α per speaker. Vocal tract length normalisation can be applied in testing as well as in training the acoustic models.

Chapter 3

Data specification

3.1 Test data

We received sets of records of 994 different words, each set was recorded by one of 12 speakers in office environment with various types of microphones. The words were saved in wav format 44,1 kHz and named according to spoken word. In each set, long and short words are included. There are 8 women and 4 men among speakers, one of them is native speaker, the others are czech. By the listening to some speech samples, we summarise the quality of the records in following table:

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

Table 3.1: Speech sets characteristics

3.2 Models

3.2.1 8 kHz models

Models are represented by context dependent phonemes. They were trained on 270 hours of speech data (American English). Models were adapted by 70 hours of meeting speech data, recorded in International Computer Science Institute (ICSI) in Berkeley.

3.2.2 16 kHz models

16 kHz models are trained on 100 hours of meeting speech data. There are 3 states per phoneme and 29 935 context dependent phonemes. Two types of 16 kHz models were available. 16 kHz PURE models are models without any speaker adaptation technique. 16 kHz + VTLN are 16 kHz models with Vocal tract length normalisation.

	8 kHz	16 kHz PURE	16 kHz + VTLN
Feature size	39	39	39
Number of gaussian components	16	16	16
Number of states	3489	7595	4787
Total parameters	4 354 272	9 478 560	5 974 176

Table 3.2: Models description

3.3 Dictionary

Dictionary was provided together with records, there are included the same words, which were recorded by speakers. It contains the word in English and it´s phonetic transcription. This transcription is realized by IPA (International Phonetic Alphabet) [2]

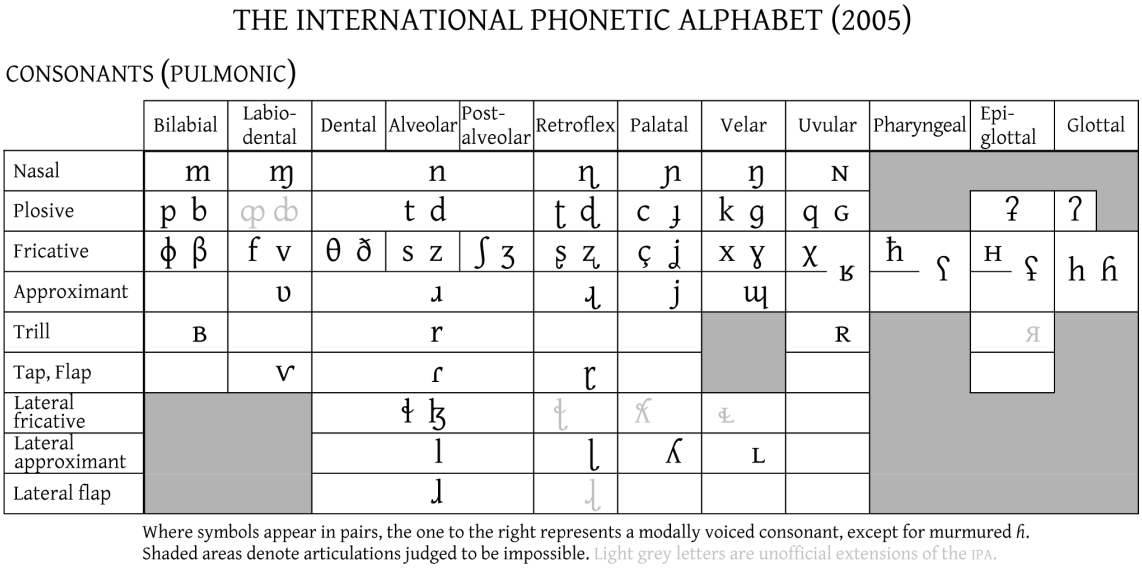


Figure 3.1: Example of IPA consonants

spacecraft: (’speIs,krA:ft)
aid: (eId)
Christ: (kraIst)
arc: (A:k)
controversial: (,kQntrE’vF:SEl)
usury: (’ju:ZErI)
railcard: (’reIl,kA:d)
fort: (fO:t)
overheat: (,EUvE’hi:t)
larva: (’lA:vE)
slowdown: (’slEU,daUn)

Figure 3.2: Part of dictionary

Chapter 4

The recognizer

The recognizer is built on HTK (Hidden Markov Model Toolkit).

The Hidden Markov Model Toolkit (HTK) [1] is a portable toolkit for building and manipulating hidden Markov models. HTK is primarily used for speech recognition research although it has been used for numerous other applications including research into speech synthesis, character recognition and DNA sequencing.

HTK consists of a set of library modules and tools available in C source form. The tools provide sophisticated facilities for speech analysis, HMM training, testing and results analysis. The software supports HMMs using both continuous density mixture Gaussians and discrete distributions and can be used to build complex HMM systems.

4.1 Phonemes conversion

The recognizer uses different phonemes set than Lingea. This section describes creation of converting table between these two sets. The difference between the phoneme notations is shown in Fig. 4.1.

LINGEA	RECOGNIZER
cognitive: (’kQgnItIv)	COGNITIVE k aa g n ih t ih v
reap: (ri:p)	REAP r iy p
hyperbole: (haI’pF:bElI)	HYPERBOLE hh ay p er b ax l ih
sight: (saIt)	SIGHT s ay t
crystallize: (’krIstE,laIz)	CRYSTALLIZE k r ih s t ax l ay z
torn: (tO:n)	TORN t ao r n
democracy: (dI’mQkrEsI)	DEMOCRACY d ih m aa k r ax s ih
broil: (brOIl)	BROIL b r oy l
madhouse: (’mHd,haUs)	MADHOUSE m ae d hh aw s
juice: (dZu:s)	JUICE jh uw s

Table 4.1: Words in different notations of IPA

In the process of creating converter between the sets, it was necessary to map individual phonemes and eventually solve ambiguities.

- **IPA phonemes set (47 phonemes)**

  A:, H, Q, V, O:, aU, aI, E, i, I, Er, b, tS, d, D, Jl, Em, En, Jn, F:, eI, f, g, h, i:, dZ, k, l, m, n, N, EU, OI, p, r, s, S, t, T, U, u:, ju:, v, w, j, z, Z

- **AMI phonemes set (44 phonemes)**

  aa, ae, ah, ao, aw, ax, axr, ay, b, ch, d, dh, eh, el, em, en, er, ey, f, g, hh, ih, iy, jh, k, l, m, n, ng, ow, oy, p, r, s, sh, t, th, uh, uw, v, w, y, z, zh,

Building conversion table

Conversion table between phoneme alphabets was created by comparison of both sets. The table contains three columns: Lingea phoneme, recognizer phoneme and alternative recognizer phonemes. According to this table, conversion script was made. To choose the best variant from phoneme alternatives, comparison script and converting script were used.

LINGEA	RECOGNIZER	variants	LINGEA	RECOGNIZER	variants
A:	ae	aa	i:	iy	
H	ae	aa	dZ	jh	
Q	aa	ao	k	k	
V	ah		l	l	
O:	ao r	ao	m	m	
aU	aw		n	n	
aI	ay		N	ng	
E	ax	axr, r	EU	ow	
i	eh	ax	OI	oy	
I	ih	ax, iy	p	p	
Er	axr	r	r	r	
b	b		s	s	
tS	ch		S	sh	
d	d		t	t	
D	dh		T	th	
Jl	el		U	uh	uw
Em	em		u:	uw	
En	en		ju:	uw	
Jn	en		v	v	
F:	er		w	w	
eI	ey		j	y	
f	f		z	z	
g	g		Z	zh	
h	hh				

Table 4.2: Phonemes conversion table

Components for testing

- **Conversion script** – Converts files from one IPA notation to another
- **Big dictionary** – 50 000 word dictionary with notation, that uses the recognizer

- **Lingea dictionary** – Dictionary to convert (Fig. 3.2)
- **Recognizer** – The base recognizer with 8 kHz models, used for testing purposes

Text testing

With help of conversion script, the sample file from Lingea was converted and compared with big reference dictionary. The comparison script returned the difference between files in %. After that mapping of one ambiguous phoneme in the conversion script was changed and the whole procedure was repeated. By ambiguous phonemes, the highest value in % was choosen. The result of this process was conversion table with theoretically best performance.

Testing with recognizer

After gaining better conversion table only with a few ambiguities, the recognition test was done. The dictionary for the recognizer was converted by conversion script from Lingea dictionary. The accuracy of the recognizer with different variants of the dictionary was noted. The final conversion table was made from the dictionary variant giving the best results.

	reference	I -> (IH -> IY)		Er -> (r -> axr)		O: -> (ao r -> ao)	
01b	41.85	44.16	2.31	41.35	-0.50	41.75	-0.10
02a	34.61	33.70	-0.91	34.10	-0.51	33.90	-0.71
05a	66.40	67.10	0.70	65.50	0.10	64.59	-1.81
07a	65.39	66.10	0.71	65.19	-0.20	63.78	-1.61
08a	38.23	34.31	-3.92	38.13	-0.10	36.62	-1.61
09a	61.87	62.07	0.20	61.47	-0.40	60.06	-1.81
10a	73.24	71.33	-1.91	73.54	0.30	71.43	-1.81
11a	54.08	54.73	0.65	54.28	0.20	52.37	-1.71
12a	51.71	50.70	-1.01	51.91	0.20	49.20	-2.51

Table 4.3: Test results - recognizer accuracy in %

4.2 Downsampling

Because testing data were available only in CD quality (44.1 kHz) and the models were trained with either 8 kHz or 16 kHz data, it was necessary to downsample the test data. Sox (Sound eXchange) was used for downsampling. Sox contains three different sample-rate conversion algorithms [5]:

1. Linear Interpolation: missing samples are linearly interpolated from existing samples.
2. Band-Limited Interpolation: missing samples are interpolated using a Kaiser-windowed sinc function from existing samples.
3. Polyphase Filtering: traditional DSP interpolation and decimation mechanisms implemented efficiently.

The polyphase filter has the best interpolation performance [5], closely followed by the bandlimited interpolation, both of them leaving linear interpolation far behind. Our only concern was signal quality, that's why the polyphase effect was chosen.

Files were downsampled with this command:

```
sox -t wav -r 44100 -s -w $in -w -r 8000 -s -t wav $out polyphase
```

- -t = filetype
- -r = rate
- -s = The sample data encoding is signed linear
- -w = The sample data size is in 16-bit words
- $in = Name of file, we want to downsample
- $out = Name of downsampled file

4.3 PLP features

PLP features are generated by this command:

```
HCopy -C $HCopyConfig -S  $ScpName;
```

- -C $HCopyConfig = config for PLP
- -S $ScpName = path to wav files, we want to program

4.4 Grammar

Grammar provides word sequence, which can the recognizer recognize. For recognition of isolated words it is in this form:

```
sil WORD sil
```

where `WORD` is parallel connection of all words in the dictionary.

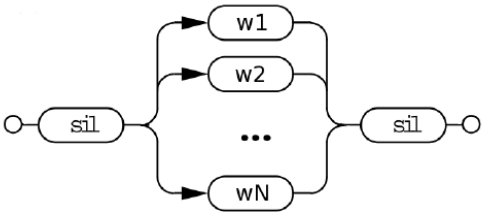


Figure 4.1: Grammar network

4.5 Recognition

Recognition was done by this command:

```
HVite -i $mlf -S $list -p $WordInsertPenalty -s $GrammarScale
-H $models -w $gram -C $config -n 20 5 $dict $model_list;
```

- -i $mlf = File with results in MLF (Master Label File) format
- -S $list = List of PLP features
- -p $WordInsertPenalty = Word insert panalty -10.0
- -s $GrammarScale = Grammar scale 1.0
- -H $models = 8 kHz or 16 kHz models
- -w $gram = Grammar
- -C $config = Config for PLP and normalisation
- -n 20 5 = N-best recognition
- $dict = Dictionary
- $model_list = Model list

4.6 Results of recognition

Results of recognition were obtained by this command:

```
HResults -d $X -I $reference $words $results
```

- -d $X = X-best recognition
- -I $reference = reference results file in mlf format (100% accuracy)
- $words = words to recognize
- $results = results file in mlf format

Chapter 5

Experiments

After creating recognizer, it was experimented with various models (8 kHz , 16 kHz, 16 kHz + VTLN). Accuracies of particular systems were noted into tables.

Accuracy was analysed on several levels:

- 1-best: recognized word equals to reference word.
- X-best: recognized word occurs in list of X best variants. (X was 5, 10, 15 or 20 in these experiments)

5.1 8 kHz results

The recognizer built with 8 kHz models was used as base system. It served for testing purposes during evaluating of conversion table between phoneme sets and also for testing executing scripts, which were with small modifications used further. 8 kHz models are quality, trained on big amount of data.

SET	1-best	5-best	10-best
01a	30.28	48.39	56.34
01b	41.85	64.29	73.24
02a	34.61	54.43	61.07
04a	22.74	40.14	47.18
05a	66.40	84.51	89.13
06a	15.90	29.07	35.71
07a	65.39	80.68	86.12
08a	38.23	56.44	63.88
09a	61.87	80.58	84.81
10a	73.24	86.72	90.04
11a	54.08	76.13	82.18
12a	51.71	69.92	78.37
average	47.82	65.72	71.98

Table 5.1: 8 kHz results - accuracy in %, measured ratio of recognized words to all words

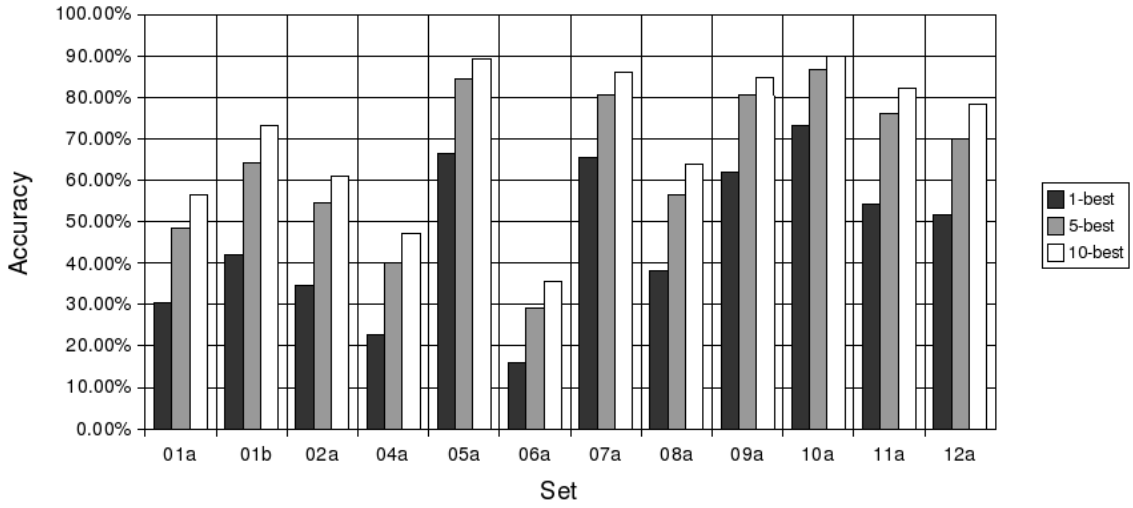


Figure 5.1: 8 kHz results - accuracy in %, measured ratio of recognized words to all words

5.2 16 kHz results

The recognizer built with 16 kHz models has better results than the recognizer with 8 kHz models. Higher sample rate leads to higher word accuracy. Difference between 8 kHz and 16 kHz 10-best accuracy average is 7%.

SET	1-best	5-best	10-best	15-best	20-best
01a	38.49	61.51	69.95	74.37	77.19
01b	57.39	78.09	84.22	87.44	89.85
02a	46.53	70.15	77.09	80.20	82.41
04a	26.43	47.44	55.58	60.40	62.61
05a	65.93	85.03	89.35	90.85	91.96
06a	28.04	44.72	52.06	56.98	60.90
07a	72.46	86.73	90.45	91.76	92.16
08a	42.01	63.92	70.35	73.57	75.48
09a	65.33	82.51	88.34	91.06	92.46
10a	78.47	91.95	94.27	94.97	95.57
11a	66.26	84.69	89.73	92.35	93.35
12a	61.21	80.50	86.13	89.35	91.76
average	54.05	73.10	78.96	81.94	83.81

Table 5.2: 16 kHz results - accuracy in %, measured ratio of recognized words to all words

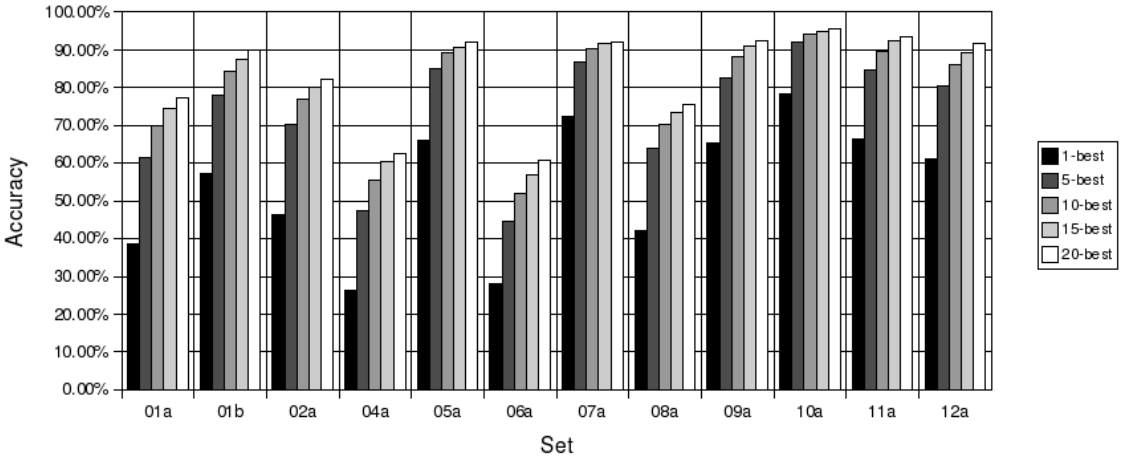


Figure 5.2: 16 kHz results - accuracy in %, measured ratio of recognized words to all words

5.3 VTLN results

For experiments with VTLN, it was necessary to find optimal normalisation factor for each speaker. Normalisation factor was searched on interval between 0.8 and 1.2. This range was divided in 100 segments of the same length. The forced alignment was done in each factor and the log likelihood of models computed. The optimum normalisation factor for given speaker is one giving the highest log likelihood. Table with optimum normalisation factors:

SET	WARPFREQ
01a	0.888
01b	0.864
02a	0.848
04a	0.848
05a	1.016
06a	0.876
07a	0.964
08a	0.912
09a	0.996
10a	1.048
11a	0.928
12a	0.996

Table 5.3: Best normalisation factors

Speaker adaptation techniques leads generally to higher accuracy rates. It may be beneficial to use these techniques, if appropriate data are available. With this system, the best results were reached.

SET	1-best	5-best	10-best	15-best	20-best
01a	50.45	71.66	78.19	81.61	84.72
01b	68.04	88.34	92.66	94.57	95.18
02a	64.82	84.62	89.15	91.26	92.26
04a	35.38	56.58	63.72	67.14	69.85
05a	69.85	88.34	92.36	93.67	94.47
06a	31.06	48.94	56.98	62.21	65.73
07a	74.57	88.44	91.86	93.07	93.47
08a	39.80	59.20	65.33	69.95	73.27
09a	67.44	83.82	88.44	91.56	93.37
10a	82.49	93.16	95.47	96.18	96.78
11a	69.92	86.32	91.05	92.86	93.36
12a	64.72	82.51	87.94	90.85	93.07
average	59.88	77.66	82.76	85.41	87.13

Table 5.4: 16 kHz + VTLN results - accuracy in %, measured ratio of recognized words to all words

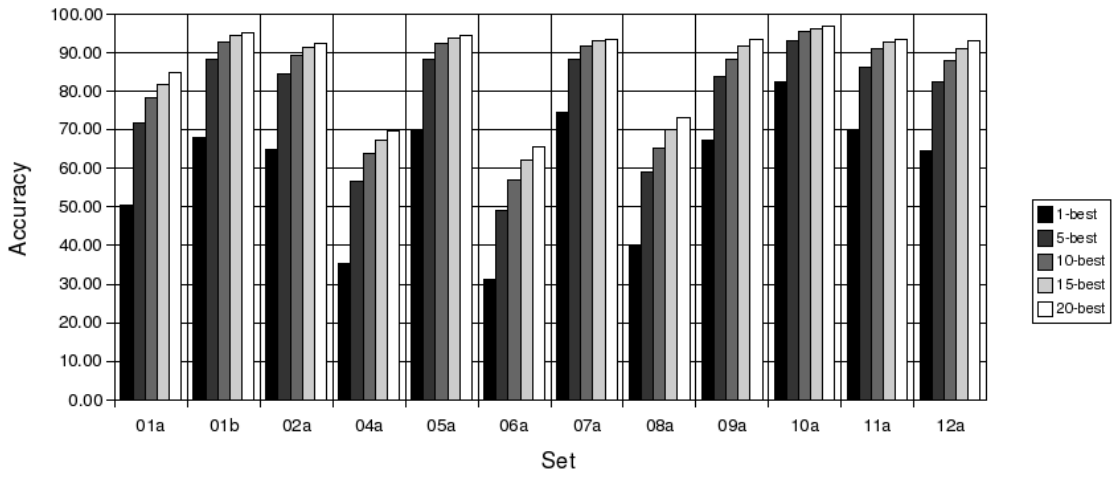


Figure 5.3: 16 kHz results + VTLN - accuracy in %, measured ratio of recognized words to all words

Chapter 6

Conclusion

The main aim of this work was to find out, if it is possible to adapt user data to existing models, which were trained in different environment. The base 8 kHz system has 65.7% word accuracy with 5-best and 72% word accuracy with 10-best. These values are sufficient for 8 kHz system. The recognizer with 16 kHz models reached 73% accuracy with 5-best (which is more than 8 kHz accuracy with 10-best) and 79% with 10-best. In higher variants of X-best, there are no such differences between accuracy results. 15-best and 20-best results differs only by 2%. After using Vocal tract length normalisation (VTLN) speaker adaptation techniques, the results were further improved. The system with 16 kHz models and VTLN reached 78% accuracy with 5-best and 83% with 10-best. With this models and 20-best, the best score 87% accuracy was reached. 83% accuracy with 16 kHz models and VTLN is sufficient value for the recognizer of isolated words, this shows that user data from office environment can be successfuly adapted to existing models.

Next work consists in use of 16 kHz models with VTLN + HLDA (Heteroscedastic Linear Discriminant Analysis). The dictionary will be increased to thousands of words with a few pronunciation variants. We also expect new datasets from the project partner. Thereby recognizing will be more difficult and the accuracy of recognizing may slightly decrease. N-best variants can contain the correct word, despite of occurrence of very similar words in the dictionary.

Bibliography

[1] Htk speech recognition toolkit [online]. [cit. 2007-05-11].
URL: <http://htk.eng.cam.ac.uk/>.

[2] International phonetic alphabet [online]. [cit. 2007-05-12].
URL: <http://en.wikipedia.org/wiki/Ipa>.

[3] B. Gold and N. Morgan. *Speech and audio signal processing: processing and perception of speech and music*. John Wiley & Sons, Inc., 2000. ISBN 0-471-35154-7.

[4] H. Hermansky. Perceptual linear predictive (PLP) analysis for the speech. *J. Acous. Soc. Am.*, pages 1738–1752, 1990.

[5] A. Wilde. Audio file sample rate conversion [online]. [cit. 2007-05-11].
URL: <http://www.leute.server.de/wilde/resample.html>.

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

Appendix A

To this bachelor work belongs CD which contains:

- technical report of this bachelor work
- 8 kHz and 16 kHz models
- dictionary with phonetic notation
- executable scripts
- demo of the recognizer