**KTH Computer Science
and Communication**

Developing acoustics models for automatic speech recognition

GIAMPIERO SALVI

Master's Thesis at TMH
Supervisor: Håkan Melin
Examiner: Rolf Carlson

TRITA xxx yyyy-nn

Abstract

This thesis is concerned with automatic continuous speech recognition using trainable systems. The aim of this work is to build acoustic models for spoken Swedish. This is done employing hidden Markov models and using the SpeechDat database to train their parameters. Acoustic modeling has been worked out at a phonetic level, allowing general speech recognition applications, even though a simplified task (digits and natural number recognition) has been considered for model evaluation. Different kinds of phone models have been tested, including context independent models and two variations of context dependent models. Furthermore many experiments have been done with bigram language models to tune some of the system parameters. System performance over various speaker subsets with different sex, age and dialect is also examined. Results are compared to previous work showing a remarkable improvement.

Preface

This Master of Science thesis was done at the Department of Speech, Music and Hearing (Tal, musik och hörsel) at the Royal Institute of Technology (KTH), Stockholm. It is fruit of a collaboration between KTH and Universitá "La Sapienza", Rome. Supervisors were Prof. Gabriella Di Benedetto, Prof. Rolf Carlson and Håkan Melin.

The thesis was later published in a summarised version on The European Student Journal of Language and Speech (WEB-SLS)

http://www.essex.ac.uk/web-sls/

Acknowledgments

There are many people to thank in connection with this work. First I begin by thanking Prof. Maria Gabriella Di Benedetto and Prof. Rolf Carlson for allowing this instructive experience and Prof. Rolf Carlson for hosting me in the TMH department. Then I thank Håkan Melin for his patience and his readiness in helping me for the whole period I have been working on this subject. Other people which indubitably deserve thanks are Kjell Elenius for everything concerned with the SpeechDat project and Kåre Sjölander for providing information about his previous work. The feeling of welcome I got from all the people working at the department has made this period particularly enjoyable, helping the development of this work: special thanks go to Philippe for being Philippe, to Thomas and Sasha for their non-stop sports and entertainments organization, Leonardo, Roberto and Vittorio for their sympathy and Peta for her humour.

Contents

Chapter 1

Introduction

1.1 Introduction

The field of speech signal analysis has been in the center of attention for many years because of the interest it is able to generate in the scientists and because of the many possible applications. These applications, related mostly to telecommunication problems, have as a goal the possibility for human beings to exchange information using the most natural and efficient means they know: speech. In this context the speech recognition enterprise is probably the most ambitious. Its goal is to build "intelligent" machines that can "hear" and "understand" spoken information, in spite of the natural ambiguity and complexity of natural languages. In thirty years, improvements that could not even be thought of before have been worked out, but still the objective of a robust machine, able to recognize different speakers in different situations, is far to be reached. The difficulty of the problem increases if we try to build systems for a large set of speakers and for a generic task (large vocabulary).

This thesis describes an attempt to build robust speaker-independent acoustic models for spoken Swedish. A collection of utterance spoken by 1000 speakers (the SpeechDat database) has been used as a statistical base from which the models have been developed. The recognition task considered includes a small vocabulary (86 words), but continuous speech is accepted. Furthermore the model structure is chosen with regard to the possibility to apply these models in different contexts and tasks. For this reason models have been tested on another database (from the Waxholm project) to evaluate their flexibility. Different applications are possible for this kind of models: they can be employed in the recognition part of complex dialogue systems, or in speaker verification systems for example.

This report contains a documentation of the steps that lead to the creation and development of the acoustic models, but also a thorough description of the system employed in the recognition phase.

1.2 Automatic Speech Recognizer

An Automatic Speech Recognizer (ASR) is a system whose task is to convert an acoustic speech signal into a symbolic description of the message coded in that signal during speech production. Two main aspects make this process particularly difficult: the "modulation" process (speech production) involves a long list of physical and psychological phenomena regarding the speaker articulatory apparatus, the phonemic system, the prosodic system, the lexicon, the syntax, the rules of discourse, the semantics, the psychology of the speaker and so on, which makes the speech signal extremely changeable with different situations and deeply coded. Second the "demodulation" process (speech recognition) involves a large reduction of information from the sampled acoustic signal (for example 8bit, 8kHz), to the text (around 5bytes/sec).

Speech recognition has, hence, an interdisciplinary nature involving many disciplines such as: signal processing, physics (acoustic), pattern recognition, communication and information theory, linguistics, physiology, computer science and psychology. Successful speech recognition systems may require knowledge on all these topics. A way to give an organisation to this knowledge is to refer to an abstract model of natural languages called Peirce's model (Hartshorne and Weiss, 1935). This model explains well the structure of our system, even though it is considered to be too simplified if the attempt is to describe natural languages.

1.2.1 Peirce's Model

Peirce's model provides a general definition of the linguistic constraints. Four components of the natural language code are included: symbolic, grammatical, semantic and pragmatic.

Symbols are the most fundamental units in a language: they might be words, phonemes, or, in written form, the alphabetic symbols.

The *grammar* of a language is concerned with how symbols are related to one another to form words or sentences. In the case that fundamental symbols are sub word units, we call *lexical* constraints the way these units are connected to form words, while *syntactic* constraints rule the way words form sentences. Both are part of the grammar.

Semantics is concerned with the way in which symbols are combined to form meaningful communication: a sentence can be grammatically correct, but be without meaning.

Finally the *pragmatic* component of the language model is concerned with the relationship of the symbols to their users and the environment of the discourse.

Knowledge at all linguistic levels is required if our aim is to build a system that can be used in different environments and for different tasks. Nevertheless the highest levels (semantics and pragmatic) are very difficult to formalize (at least in conventional engineering ways) and they are usually considered as belonging to the field of artificial intelligence. The usual way to overcome this problem is to restrict

the field of application of the system to be built and to use in its construction only knowledge that can be formalized in a mathematical way.

In Figure 1.1 a schematic representation of Peirce's model is depicted.
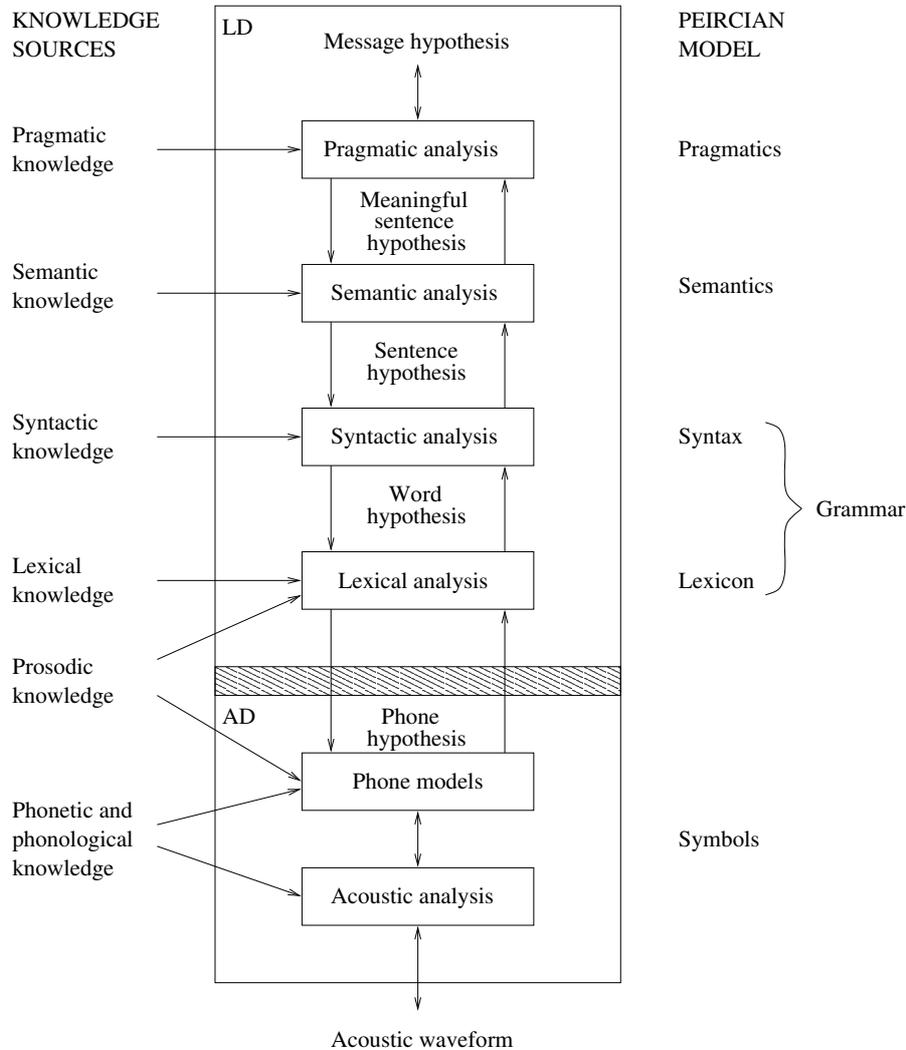
Figure 1.1. A schematic representation of the Peirce's model for natural languages.

This work is focused on the acoustic part of a speech recognizer and hence will deal essentially with the first two levels of language constraints.

There are two different approaches in the way knowledge is obtained and used: the first is the *deterministic* approach: all the phenomena involved in speech production are analyzed and formalised from a physical and physiological point of view (when possible). The utterance to be recognised is then compared to a synthetic signal generated according to some acoustic models developed on this physical and

physiological base. The computational requirements needed in this case are low, but a big effort is required in formalizing the basic aspects of speech production.

The second approach (*stochastic*) is based on statistical methods: some opportune models are "trained" on a large amount of data in order to estimate their parameters and to make them fit to the problem of interest. This second approach requires a big computing power and large data bases.

The rapid development of computers in the last years has made the statistical approach more and more attractive, but knowledge on speech production at a physical level is still required. First of all because a long signal processing chain is indispensable in order to extract the features that best represent the speech signal characteristics and to reduce the amount of data to deal with per unit of time. Second of all because the choice of model structures must always be motivated by an analogy with the physical phenomena those models try to imitate.

1.3 Structure of the work

This thesis is structured as follows. The acoustic models are created on a phonetic base hence a description of the Swedish phonology is reported in Chapter 2. Chapter 3 describes the SpeechDat database used in model building and testing. Chapter 4 describes the ASR structure. Chapter 5 describe the mathematical theory upon which model building is based. In Chapter 6 all the steps in model building and developing are documented, and finally results are presented in Chapter 7 and conclusions in Chapter 8.

Chapter 2

Swedish Phonology

As a first topic of this thesis, the Swedish phonology is described. The aim of this work is to build acoustic models for different speech features, and hence the Swedish language characteristics are an important background to understand the following. Section 2.1 describes the Swedish phonemes, while Section 2.2 describes the main characteristics of pronunciation in different dialect areas.

2.1 Swedish Phonemes

In the Swedish language, sounds can be classified into 46 phones. 23 of these are vowels, while the other 23 are consonantal sounds. Among the consonants there are 4 nasals, 7 fricatives, 3 liquids, 8 plosives, 1 h-sound (5 retroflex allophones are included in their respective groups).

2.1.1 Swedish Vowels

The Swedish written language has 9 vowels [5] (a, e, i, o, u, y, ä, ö, å) which, when spoken, can occur in a *short* or *long* version according to some pronunciation rules ("A long vowel in a stressed position can only occur before a short consonant and vice versa. In unstressed position there is no distinction between long and short vowels..." [5]). An example is in table 2.1. Most of them are front vowels (e, i, y, ä, u) which makes their distinction difficult for an untrained ear.

In figure 2.1 each vowel occupy a point in the F1/F2 plain, where F1 and F2 are respectively the first and second formants. Formant data has been taken from Fant (1973) [5], regarding seven female speakers. The X and Y axes showed in the figure provide a way to relate formant variations to mouth configurations. Moving along the X axis, the tongue position is approximately constant, while moving along the Y axis mouth opening is constant.

2.1.2 Consonants

Swedish consonants are listed in Table 2.2 in the STA symbols. In the table, ac-

vowel	long			short		
	STA	SAMPA	example	STA	SAMPA	example
a	A:	A:	hal ()	A	a	hall ()
e	E:	e:	vet (know)	E	e	sett (seen)
i	I:	i:	bil (car)	I	I	till (to)
o	O:	u:	rot (root)	O	U	bott (lived)
u	U:	}:	mun (mouth)	U	u0	buss (bus)
y	Y:	y:	lys (shine)	Y	Y	ryss (russian)
ä	Ä:	E:	lät (sounded)	Ä	E	lätt (easy)
ö	Ö:	2:	nöt (beaf)	Ö	2	kött (meat)
å	Å:	o:	kål (cabbage)	Å	O	vått (wet)
pre-r	long			short		
allophone	STA	SAMPA	example	STA	SAMPA	example
är	Ä3	{:	här			
ör	Ö3	9:	för			
err				Ä4	{	herr
örr				Ö4	9	förr
schwa vowel	long			short		
allophone	STA	SAMPA	example	STA	SAMPA	example
e	E0	@	pojken			

Table 2.1. "Long" and "short" vowels in Swedish phonetics

cording to Fant [5], phones articulated with a palatal or velar tongue position are placed at the bottom, except for the liquids and the **H** phone. The labial member of each group is placed to the left, and the dental to the right

Nasals

Nasal sounds (**M**, **N**, **2N**, **NG**) are produced with the vocal tract totally constricted at some point in the mouth. The sound is hence radiated by the nasal tract, while the oral tract behaves as a resonant cavity which traps acoustic energy and introduces zeros in the transfer function of sound transmission at some natural frequencies. Nasals can be distinguished by the place at which the constriction of the oral tract is made. For **M** the constriction is at the lips, for **N** it is behind the teeth, for **NG** it is forward of the velum. When uttering **2N** the position of the tongue is similar to that for **N**, but it is curled.

Fricatives

Fricative sounds can be divided into voiced and unvoiced sounds. There are four unvoiced fricatives in Swedish: **F**, **S**, **TJ**, **SJ** of which **TJ** occurs only in word initial position. Unvoiced fricative sounds are produced by exciting the vocal tract by a steady air flow which becomes turbulent in the region of a constriction in the vocal tract.

Voiced fricatives are **V** and **J**. **V** is produced as **F**, but with excitation of the vocal cords. **J** may also be pronounced as half-vowel.
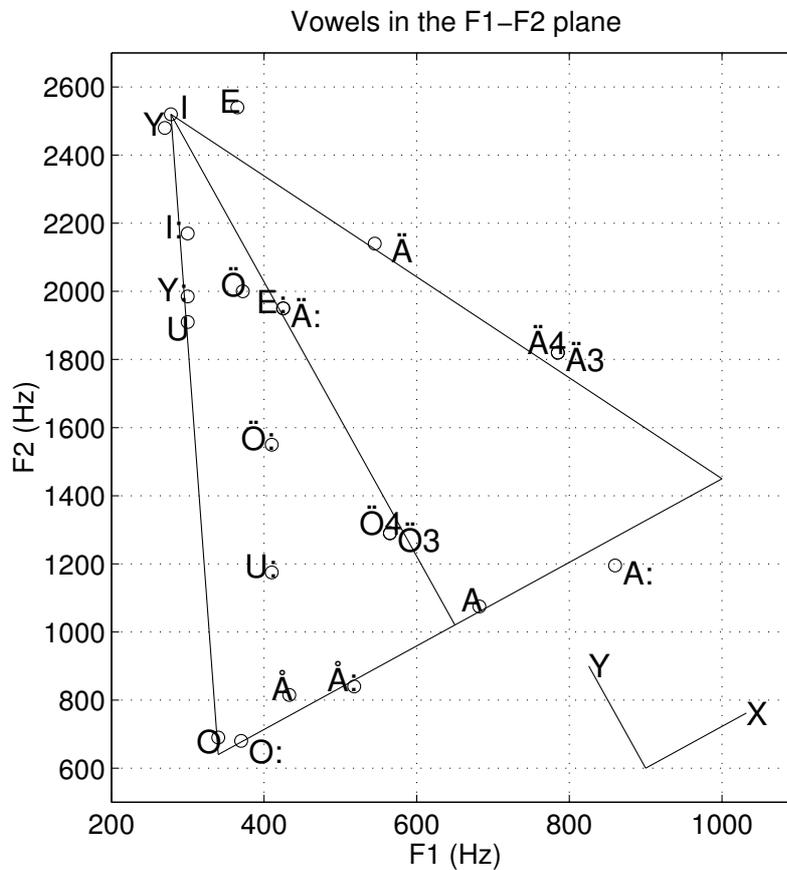
Figure 2.1. Vowel distribution in the F1-F2 plane

Liquids

The **R** sound can vary depending on the speaker. It can sound as a trill or an uvular fricative in the south of Sweden, or as a retroflex voiced fricative in the area around Stockholm. **L** is also classified as lateral because it is produced rising the tongue. **2L** is a variant of **L** "produced with retracted tongue position and uvular constriction" [5].

Plosives (Stops)

These sounds are produced by changing the vocal tract configuration. They are normally classified as non-continuant sounds. All stops have a first period in which pressure is built up behind a total constriction in the oral tract. Then the pressure is suddenly released (second period). To each stop corresponds a different constriction position.

A distinction can be made between unvoiced stops **K**, **P**, **T**, **2T** (no sound is

```
   h      Liquids     Voiced fricative   Unvoiced fricative         Nasals
   H            L    V                    F              S    M                    N
          2L                                       SJ                        2N
        R                          J          TJ                      NG
                        Voiced stops       Unvoiced stops
                        B            D    P              T
                              2D                 2T
                        G                   K
```

Table 2.2. Swedish consonants

produced during the first period) and voiced stops **B**, **D**, **2D**, **G** (a small amount of low frequency energy is radiated through the walls of the throat).

H-sound

The h-sound (**H**) is produced by exciting the vocal tract by a steady air flow, without the vocal cords vibrating, but with turbulent flow being produced at the glottis.

Retroflex Allophones

Retroflex allophones (**2S**, **2T**, **2D**, **2N**, **2L**) have already been mentioned in their groups. They are now grouped together because they will be object of discussion in the following of this work. These symbols apply to alveolar articulation of sounds spelled r plus s, t, d, n, or l and occur only in positions after a vowel. The distinction between the retroflex version of a phoneme and the normal one (for example **2L** and **L**) is not important in the southern dialects of Sweden.

2.2   Dialectal Variations

Swedish pronunciation depends strongly on the origin of the speaker, while STA symbols are independent of the specific dialect sound qualities. For this reason in this section some variations from the standard pronunciation rules are listed for the seven major Swedish dialect regions [4]. Most of differences regard the pronunciation of vowels.

1. *South Swedish*
   South Swedish diphthongization (raising of the tongue, late beset rounding of the long vowels), retracted pronunciation of **R**, no supra-dentals, retracted pronunciation of the fricative **SJ**-sound. A tense, creaky voice quality can be found in large parts of Småland.

2. *Gothenburg, west, and middle Swedish*
   Open long and short **Ä** and (sometimes) **Ö** vowels (no extra opening before **R**), retracted pronunciation of the fricative **SJ**-sound, open **Å**-sound and **L**-sound.

3. *East, middle Swedish*
   Diphthongization into **E/Ä** in long vowels (possibly with a laryngeal gesture), short **E** and **Ä** collapses into a single vowel, open variants of **Ä** and **Ö** before **R** (Ä3, Ä4 and Ö3, Ö4 in STA).

4. *Swedish as spoken in Gotland*
   Secondary Gotland diphthongization, long **O**-vowel pronounced as **Å**.

5. *Swedish as spoken in Bergslagen*
   U pronounced as central vowel, acute accent in many connected words.

6. *Swedish as spoken in Norrland*
   No diphthongization of long vowels, some parts have a short **U** pronounced with a retracted pronunciation, thick **L**-sound, sometimes the main emphasis of connected words is moved to the right.

7. *Swedish as spoken in Finland*
   Special pronunciation of **U**-vowels and long **A**, special **SJ** and **TJ**-sounds, **R** is pronounced before dentals, no grave accent.

Chapter 3

SpeechDat Database

SpeechDat [15] is a project involving 15 European countries (listed in tab. 3.1) and essentially based on the country partner's fixed telephone networks. Speakers

Country	Language (variant)	Partner	Recorded Speakers
United Kingdom	English	GEC	+4000
United Kingdom	Welsh	BT	2000
Denmark	Danish	AUC	+4000
Belgium	Flemish	L&H	1000
Belgium	Belgian French	L&H	1000
France	French	MATRA	5000
Switzerland	Swiss French	IDIAP	+2000
Switzerland	Swiss German	IDIAP	1000
Luxembourg	Luxemb. French	L&H	500
Luxembourg	Luxemb. German	L&H	500
Germany	German	SIEMENS	+4000
Greece	Greek	KNOW & UPATRAS	5000
Italy	Italian	CSELT	+3000
Portugal	Portuguese	PT	+4000
Slovenia	Slovenian	SIEMENS	1000
Spain	Spanish	UPC	+4000
Sweden	Swedish	KTH	5000
Finland	Finnish Swedish	DMI	1000
Finland	Finnish	DMI	4000
Norway	Norwegian	TELENOR	1000

Table 3.1. European countries involved in the SpeechDat project

are selected randomly within a population of interest including all possible types of speaker. Speaker characteristics considered in the corpus collection project are essentially: sex, age and accent.

3.1 Swedish Database

KTH is collecting a 5000 speaker database in Swedish as spoken in Sweden (population: 8.8 million at the end of 1995) and DMI in Finland collects a 1000 speaker

Finnish Swedish database [15]. Sweden has been divided into the dialectal areas shown in Figure 3.1 by Professor Emeritus Claes-Christian Elert, a Swedish expert in this field. This division does not regard genuine dialects, but rather the spoken

Figure 3.1. Dialectal areas in Sweden

language used by most people in the areas defined. A definition of these areas is at page 8 of the previous chapter.

3.1.1 Database Features

During this work a 1000-speaker subset of the 5000 speaker database (FDB1000 [4]) has been available. For each speaker (session) a variety of different items are provided for different tasks. Only a part of them (described below) has been used in training and testing the models.

Each utterance has been stored in a 8bit, 8kHz, A-low format, and an ASCII label file is provided for each speech file, including information about sex, age,

accent, region, environment, telephone type, and a transcription of the uttered
sentence or word. Items used in model training contain for each speaker:

- *9 phonetically rich sentences* (`S1-S9`)

- *4 phonetically rich words* (`W1-W4`)

while items for development and evaluation tests are:

- *1 sequence of 10 isolated digits*[1] (`B1`)

- *1 sheet number (5+ digits)* (`C1`)

- *1 telephone number (9-11 digits)* (`C2`)

- *1 credit card number (16 digits)* (`C3`)

- *1 PIN code (6 digits)* (`C4`)

- *1 isolated digit* (`I1`)

- *1 currency money amount* (`M1`)

- *1 natural number* (`N1`)

3.1.2 Noise Transcriptions

A lexicon file provides a pronunciation for every word in the transcriptions with
exception for the following cases which have been introduced to take into account
special noises or speaker mistakes in the database.

- *filled pause* [`fil`]: is the sound produced in case of hesitation.

- *speaker noise* [`spk`]: every time a speaker produces a sound not directly
  related to a phoneme generation, such as lip smack.

- *stationary noise* [`sta`]: is an environmental noise which extends during the
  whole utterance (white noise).

- *intermittent noise* [`int`]: is a transient environmental noise extending in a
  few milliseconds and possibly repeating more than once.

- *mispronounced word* (`*word`)

- *unintelligible speech* (`**`)

- *truncation* (`~`): ~utterance, utterance~, ~utterance~.

For all these symbols a particular model must be introduced. In section 6.2 noise
models are described.

---

[1]Swedish digits and their variants are described in appendix C

3.1.3 Speaker Subsets

In the SpeechDat documentation [2] a way to design evaluation tests is proposed. For 1000 speaker databases a set of 200 speakers should be reserved for these tests, and the other 800 speakers should be used for *training*. In our case training includes many experiments, and in order to compare them, *development tests* are needed before the final *evaluation test*. For this reason a subset of 50 speakers was extracted from the training set.

In order to maintain the same balance as the full FDB database, speakers for each subset are selected using a controlled random selection algorithm: speakers are divided into different cells with regard on region and gender. Then an opportune number of speakers is randomly selected from each cell to form the wanted subset.

3.1.4 Statistics

Results from model tests are presented as mean error rate figures over speakers in the development (50 speakers) and evaluation (200 speakers) subsets respectively. To give an idea of the consistency of these results, some statistics on the database are given in this section. In Table 3.2 speakers are divided according to gender and age, while in Table 3.3 the distinction is about dialect regions. These distinctions will be particularly useful in Chapter 7, where per speaker results are discussed.

Tables show how the balance in the full database is preserved in each subset. They also show how some groups of speakers are not well represented in the evaluation subset. For example no Finnish speaker nor speakers from Gotland are present in this subset. Furthermore speakers from Bergslagen do not represent a good statistical base, a fact to consider when discussing results. The same can be said about young and old speakers in the age distinction.

To give an idea of the consistency of results, in Chapter 7 the standard deviation is reported in addition to the mean values in the case of results for speaker subsets.

Sex	Age	Train	Development	Evaluation	tot
F	young(≤ 16)	21	1	5	27
	middle	396	26	106	528
	old(> 65)	16	1	5	22
	tot	433	28	116	577
M	young(≤ 16)	14	1	1	16
	middle	284	20	80	384
	old(> 65)	19	1	3	23
	tot	317	22	84	423

Table 3.2. Number of speakers in three age ranges for female (F) speakers and male (M) speakers respectively in the Training, Development and Evaluation subsets.

Region	Train	Development	Evaluation	tot
Bergslagen	24	3	4	31
EastMiddle	281	9	85	375
Gothenburg	131	12	29	172
Norrland	164	11	46	221
South	112	9	28	149
Finnish	3	2	-	5
Gotland	6	2	-	8
Other	29	2	8	39
Tot	750	50	200	1000

Table 3.3. Number of speakers belonging to different dialectal areas in the Training, Development and Evaluation subsets.

Chapter 4

ASR Structure

Referring to the Peirce's model of natural languages introduced in Chapter 1, the structure of the automatic speech recogniser is described in this chapter. The aim of this chapter is to define which is the knowledge involved in the system at all different levels and to describe how this knowledge is applied. Since the phoneme modelling is the target of this work, its description is developed in the next chapter.
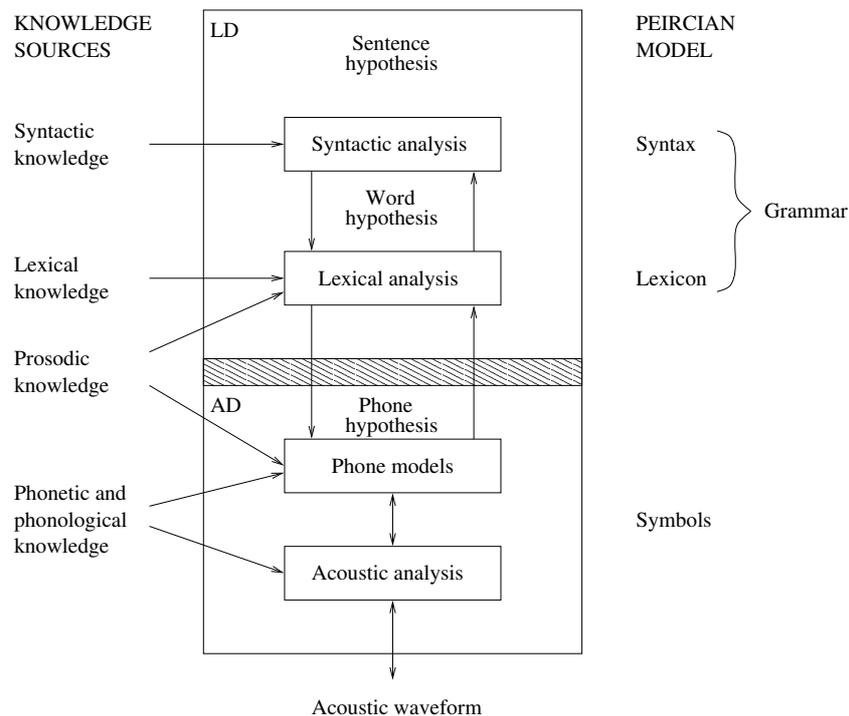
Figure 4.1. A reduced version of the Peirce's model

4.1 ASR structure

Figure 4.1 is a reproduction of Figure 1.1 in Chapter 1, with the highest levels of
linguistic constrains removed, since they don't take part of the following discussion.
The system is structured in four main blocks as depicted in Figure 4.2. The signal
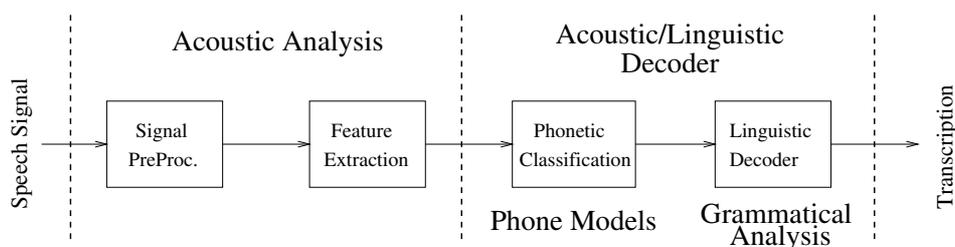
Figure 4.2. ASR structure.

is processed by each block in a linear chain, and its content changes gradually from
an acoustic to a symbolic description of the message.

The first two blocks, corresponding to the "Acoustic analysis" block in Fig-
ure 4.1, provide the acoustic observations. These observations are vectors of coeffi-
cients that are considered to represent the speech signal characteristics. The aim of
this process is not only to reduce the data flow, but also to extract information from
the signal in an efficient form for the following processing. The process is described
in section 4.2.

The last two blocks provide the *acoustic classification* and the *linguistic analysis*,
corresponding to the "Phone models" and "grammatic"levels in the Peirce's model.
Those two blocks are described separately in section 4.3 because they are based on
completely different sources of knowledge. Yet, when focusing our attention on the
implementation of the system, we will see that knowledge from the phonetic and
linguistic levels is used in a homogeneous way in the *acoustic/linguistic decoder*, and
that different sources of information are indistinguishable in the final model. Last
part of section 4.3 describes the model that implements this part.

4.2 Acoustic Analysis

Acoustic analysis is based on the speech signal characteristics. The speech signal
is a non stationary process, and thus the standard signal processing tools, such as
Fourier transform, can not be employed to analyze it. Nevertheless, an approxim-
ated model for speech production (see Appendix A) consists essentially of a slowly
time varying linear system excited either by a quasi periodic impulse train (voiced
speech) or by random noise (unvoiced speech) [14]. A short segment of speech
can hence be considered as a stationary process and can be analyzed by standard
methods. The first processing block (Section 4.2.1) consists of the computation of

the short time DFT, which is usually called in the literature "stDFT". Further-more, the short time spectrum of the speech signal contains slow variations (vocal tract filtering) combined with fast variations (glottal wave) [9]. The cepstral coeffi-cients (Section 4.2.2) allow us to separate these two components in the short time spectrum. These coefficients are employed also because they are approximately uncorrelated (compared to the linear predictive analysis) holding the maximum of information, and approximately satisfying the HMM assumptions, as we will see later.

4.2.1 Signal Processing
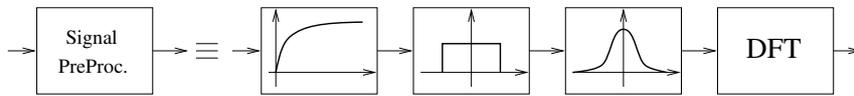
Figure 4.3. Signal processing block.

The signal pre-processing block is expanded in fig. 4.3. The pre emphasis block has a first order response function:

$$H(z) = 1 - \alpha z^{-1}$$

where in our case $\alpha = 0.97$. Its aim is to spectrally flatten the signal and to make it less susceptible to finite precision effects later in the processing chain.

The next three blocks provide a short time discrete Fourier transform. All steps are described in figure 4.4. In the first step, blocks of $N_w = 256$ samples are used together as a simple frame. Consecutive frames are overlapping and spaced of $M = 80$ samples (10ms). Than samples in the same frame are weighted by a Hamming window of width $L = 256$ whose definition is

$$w[n] = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{L-1}\right) & \text{if } 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

This is to reduce the effects of side lobes in the next short time spectrum analysis provided by the DFT block.

4.2.2 Feature Extraction

DFT block outputs a $N = 256$ frequency samples vector per frame, then a mel frequency filter-bank is used to further reduce the data flow. This non linear filter-bank was chosen to imitate the ear perceiving characteristic (see Appendix A). Then a homomorphic transformation is performed to evaluate the "cepstrum coefficients".
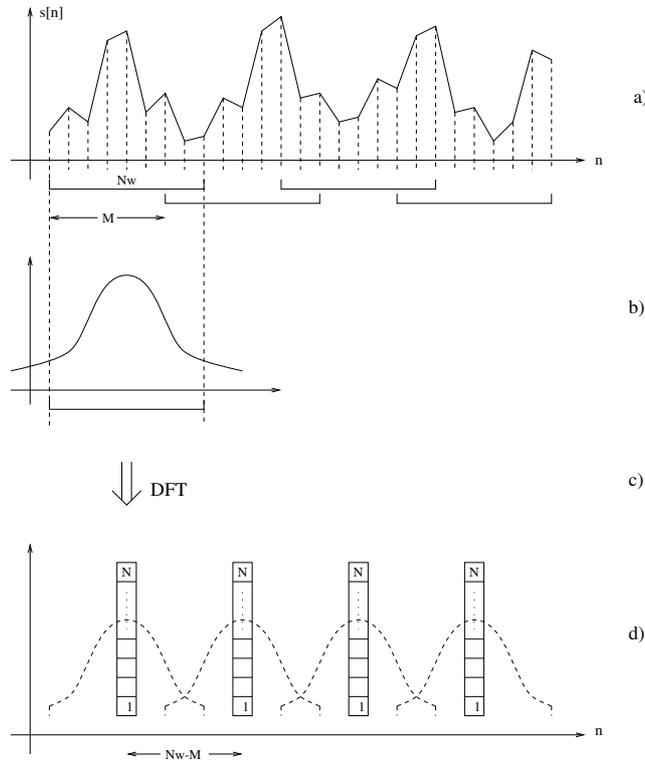
Figure 4.4. Signal processing steps

PSfrag replacements
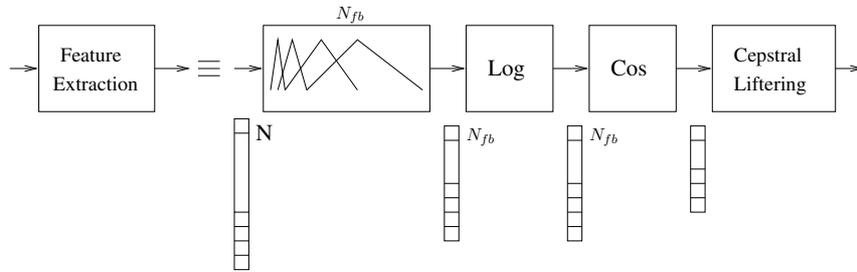
Figure 4.5. Feature extraction block.

Cepstrum analysis, based on the theory of homomorphic systems ([14]), has been proved to be a robust method in speech signal analysis. The energy coefficient is added.Then *delta* and *acceleration* coefficients are computed to keep trace of the strong time correlation in the speech signal [1].

---

[1]The application of Hidden Markov Model theory is based on the hypothesis that different frames are statistically independent

4.3 Acoustic/Linguistic Decoder

This part is the core of every ASR. Its task is to classify strings of observation frames into classes of language units. Many algorithms have been designed to solve this problem, but there are two basic classes of methods upon which almost all contemporary speech recognition systems are based. The first class contains deterministic approaches also called *dynamic time warping* algorithms. The second class contains stochastic approaches based on statistical methods. They are essentially *hidden Markov models* (HMMs), which are employed in our system, and *artificial neural networks*.

4.3.1 Phone Models

HMMs and their applications to the speech recognition problem are described in Chapter 5. At this stage we just point out that the main problem involved at this level is a *template-matching* problem. The features of the test utterance have to be aligned to those of the reference utterance, or in our case to the corresponding model states, on the base of acoustic similarities. These problems can be solved with a best path search algorithm, such as Viterbi or Forward-Backward (see chapter 5).

4.3.2 Linguistic Decoder

The linguistic decoder is responsible for the phone model concatenation in order to formulate higher level hypotheses on the spoken utterance. Since terminals in our linguistic model are sub-word units, LD is composed of two layers. The first level consists of a lexicon file in which a pronunciation (sequence of phonemes) is provided for each word involved in the recognition task. Eventually multiple pronunciations are allowed. In Swedish, for example, as described in Appendix C, many digits or numbers have more then one possible pronunciation.
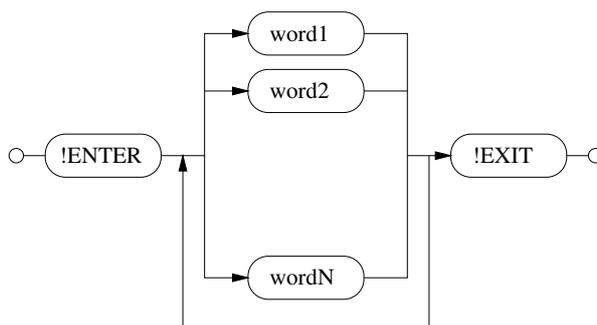
Figure 4.6. Word network

The second level generates sentence hypotheses. In the system used in this work, this level consists of a network (Figure 4.6) in which every node corresponds to a

word and every arc correspond to an allowed transition from a word to another. To each arc a probability measure is attached, representing a measure of the likelihood that the next word in that arc follows the previous word.

Researchers have investigated many methods to evaluate this likelihood. Depending on the task these methods are based on linguistic considerations that lead to rule based grammars, or on statistics on ad hoc corpora of sentences. Among the latter methods N-grams are often employed: for each sequence of N words in the dictionary, the training files in the database are scanned and the number of times that sequence appear is evaluated. Then these numbers are scaled to obtain expectation values.

In this work a bigram grammar has been employed: only couples of words are considered when computing these statistics.

Two parameters can be changed in the grammar model to tune the entire process: the first one is a grammar weight by which the arc probabilities are multiplied. This is a measure of the confidence given to the acoustic model performance. Good acoustic models need a low grammar weight. The second parameter, called insertion penalty, is introduced to find a trade off between two kinds of mistakes the system can run into: "insertions" and "deletions". To reduce the effect of those errors, a penalty can be introduced between two words: an high value for this penalty reduces the number of insertions, while a low value reduces the number of deletions. Results in development tests have been obtained after an optimization over these two parameters for every experiment, showing that the grammar weight value is essentially experiment independent, while insertion penalty value is weakly dependent on the experiment (phone model intrinsic accuracy)

4.3.3   Implementation

In the real implementation *phone models* and the different levels of the *linguistic decoder* are merged together to form a uniform system. This system is a network that is responsible for generating recognition hypotheses at all the different levels.

This merging procedure is best explained with a simple example. We consider an application in which two short words (*noll* and *ett*, "zero" and "one") are to be recognized when uttered continuously. The word *noll* is twice as probable as the other, according to some statistical measurements (grammatical model). We first describe each part of the *acoustic/linguistic decoder* separately for this example and then we put the pieces together.

phone models

As will be clear in the next chapter, a statistical model is associated to each phoneme in the task; in figure 4.7 we show the model for the phoneme **N**.

In our example the phonemes involved are **N**, **Å**, **L**, **A**, **E**, **T** and **#**, where the last takes into account short optional pauses between words in connected speech. There is a model also for silence segments that we call **sil**.

N
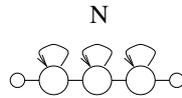
Figure 4.7. Phone model for the phoneme N

lexicon

The lexicon file specifies the pronunciations for each word:

```
noll     N Å L #
noll     N Å L A #
ett      E N #
ett      E T #
ett      E T A #
!ENTER   sil
!EXIT    sil
```

The last two entries serve as initial and final nodes in the model we are building. Each word pronunciation consists of a sequence of phonemes followed by a short pause model.
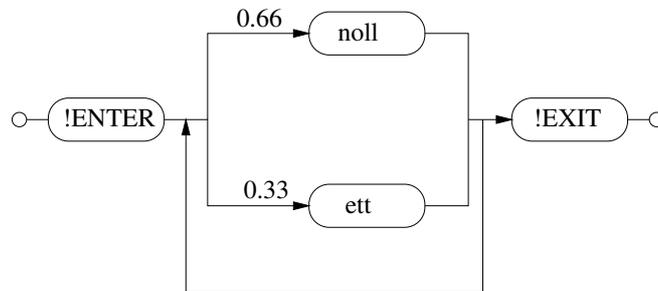
Figure 4.8. Grammar definition for the example

grammar

The grammar, defined in the example, is depicted in figure 4.8 and allows a concatenation of the two words according to transition probabilities indicated as numbers in the figure.

merging

Entries from the lexicon are first expanded into a concatenation of phone models. An example of this procedure is depicted in figure 4.9 for the first pronunciation of the word `noll`.
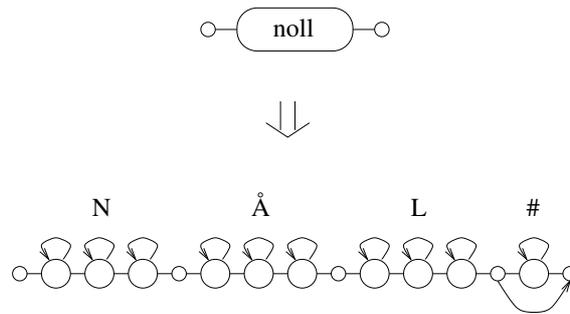
Figure 4.9. Pronunciation expantion for the word "noll"

In the second step the network specified by the grammar is considered: first each word node is cloned into a number of nodes taking into account different pronunciations. Then each of these pronunciation nodes is substituted by the concatenation of models obtained according to the lexicon. Figure 4.10 shows the final network that would be used in this simple application. We note that in this network there is no distinction between grammar, lexicon, and phonetic models: everything is modeled by arcs (possibly including transition probabilities) and nodes.
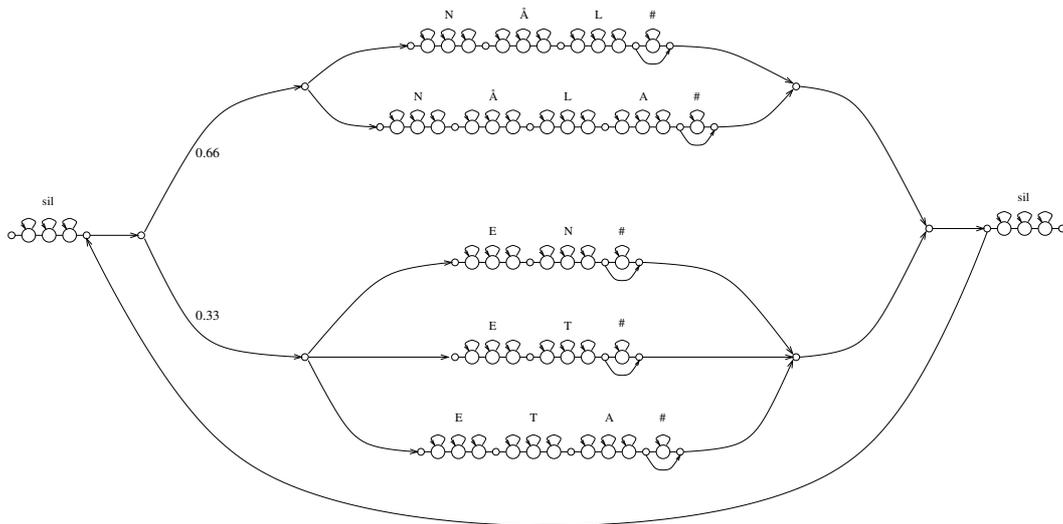
Figure 4.10. Final network that is used in recognition

Chapter 5

Hidden Markov Models

Hidden Markov models (HMM) have been successfully employed in speech recognition for many years, and several large-scale laboratory and commercial ASR systems are based on this theory. This chapter describes the HMM definition in Section 5.2 and the applications in speech recognition in Section 5.3. An HMM toolkit, described in Section 5.1 has been used to build and test the models.

5.1   The HTK Toolkit

HTK is an HMM oriented toolkit developed at the Cambridge University. It consists of a set of tools enabling the definition, initialization, re-estimation, and editing of sets of continuous mixture Gaussian HMMs. Tools to perform speech coding, alignments, model clustering and speech recognition are also included. Version 2.1.1 was used in this research [18].

5.2   Hidden Markov Models

Hidden Markov Model theory is based on doubly embedded stochastic processes. Each model can visit different states x_t and generate an output o_t in subsequent time steps. In these systems, a first stochastic process is responsible for the state transitions, while the second process generate the output depending on the actual state. This structure makes HMMs able to model phenomena in which a non observable part exists. If we define:

- a set of N reachable states $S = \{s_1, s_2, ..., s_N\}$

- a set of M possible output symbols $V = \{v_1, v_2, ..., v_M\}$[1]

a set of three probability measures, A, B, and π is required, to specify an HMM, where:

---

[1]In our application continuous output HMM are employed. In this case V can be thought of as the discrete representation of the continuous output on computing machines, and M is very large.

- $A = \{a_{ij}\}$, $a_{ij} = Prob\{x_{t+1} = s_j | x_t = s_i\}$ is the state transition probability distribution

- $B = \{b_j(k)\}$, $b_j(k) = Prob\{O_t = v_k | x_t = s_j\}, \forall k \in [1, M], \forall j \in [1, N]$ is the observation symbol probability distribution in state s_j

- $\pi = \{\pi_i\}$, $\pi_i = Prob\{x_1 = s_i\}, \forall i \in [1, N]$ is the initial state distribution

In the following we will refer to an HMM with the symbol $\lambda = (A, B, \pi)$.

HMMs employed in this work have a particular structure showed in Figure 5.1. In this *left-to-right* model, the states s_i are visited in a sequence so that the state index i is a monotone sequence of time. This means that if $x_t = s_i$ is the state visited at time t and $x_{t\prime} = s_j$ is the state visited at time $t\prime$ for all $t\prime > t$, $j \geq i$. In the figure an HTK style model is shown in which no state-to-output probability is specified for the first state (s_1) and for the last state (s_5). This states are called *non emitting* states and they are used only as entry and exit states of the model.
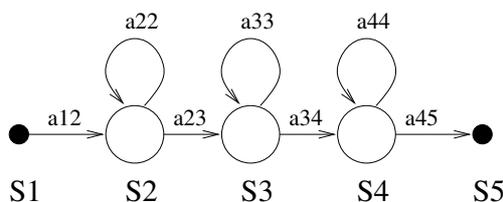
Figure 5.1. Left-to-right HMM

5.3   HMMs and ASRs

When applying HMMs to the speech recognition task three main problems have to be dealt with:

1. Given a sequence of training observations for a given word, how do we train an HMM to represent the word? (Training problem).

2. Given a trained HMM, how do we find the likelihood that it produced an incoming speech observation sequence? (Single word recognition problem).

3. Given a trained HMM, how do we find the path in the HMM state space that most likely generates an incoming speech observation sequence? (Continuous speech recognition problem)[2].

In the next sections these problems are described and applied to the speech recognition task.

---

[2]in this case the HMM is a network connecting many word level HMMs. Finding the most likely path means being able to transcribe the word sequence.

5.3.1   The Recognition Problem

The most natural measure of the likelihood of a model λ given the observation sequence O would be $Prob(\lambda|O)$. However the available data do not allow us to characterize this statistic during the training process.

A way to overcome this problem is to choose as likelihood measure $Prob(O|\lambda)$, the probability that the observation sequence O is produced by the model λ. A straightforward way to evaluate $Prob(O|\lambda)$ would be to enumerate all possible state sequences in the time interval $[1, T]$. Then $Prob(O|\lambda)$ would simply be the sum[3] of all probabilities that model λ generates the observation sequence O passing through each state sequence.

This method, even if theoretically simple, is computationally impracticable, depending on the huge number of state sequences to deal with in normal applications. Fortunately a more efficient procedure, called the Forward-Backward algorithm, exists.

The possibility for such a method lies on the fact that in the previous method many computations are repeated more than once because each state sequence is considered separately. In the Forward-Backward procedure (FB) the idea is to compute at each time step t and for each state s_i a global parameter which holds information of all possible paths leading to state s_i at time t. This parameter can be either a forward partial observation probability or a backward partial observation probability, allowing two different versions of the algorithm. Thanks to the definition of these global parameters $Prob(O|\lambda)$ can be computed recursively saving many orders of magnitude in computational time.

In continuous speech recognition the problem of model scoring is more complex [11]. Each observation sequence represents a concatenation of words. If we consider the discussion about the ASR in Chapter 4 and in particular Figure 4.6 (page 21), we see how the network depicted in that figure can be viewed as a big HMM. In the figure each word is substituted with a concatenation of models (representing its pronunciation in the case of phone models), and an additional state is appended in the end of each word representing the word end and holding the name of the word. Hence recognition results in scoring the entire network on the observation sequence with a fundamental difference with respect to the single word recognition problem. In this case the path through the network must be recorded to allow transcribing the spoken utterance.

The FB algorithm, computing the partial observation probabilities, discards this information ($P(O|\lambda)$ is computed on *any path*), and can not be used for this task.

For this reason another algorithm called the Viterbi algorithm is used instead. When used to compute $P(O|\lambda)$ this method is based on the assumption that given an observation sequence, the best path in the recognition network is far more likely than all the other paths, and hence $P(O|\lambda) \approx P(O, \delta|\lambda)$, where δ is the best path. With this assumption, in each time step, only one path per state survives, while the others are discarded. This algorithm reduces the computational time and provides

---

[3]Mutually exclusive events.

a method for reconstructing the best path in the network in a Backtracking phase. Forward-Backward and Viterbi algorithms are described analytically in Appendix B.

5.3.2 The Training Problem

The aim is to adjust the model parameters (A, B, π) to maximise the probability of the observation sequence, given the model. This is an NP-complete problem, and hence has no analytical solution. In fact, given any finite observation sequence as training data, we can only choose $\lambda = (A, B, \pi)$ such that $Prob(O|\lambda)$ is locally maximised. There are many methods for doing it, among which gradient search techniques, expectation-modification technique and genetic algorithms. Baum introduced in 1966 an estimation method based on the Forward-Backward algorithm. This method, known as Baum-Welch Re-estimation algorithm, makes use of an old set of model parameters λ to evaluate quantities that are then used to estimate a new set of model parameter, say $\hat{\lambda}$. Baum proved that two cases are possible after the re-estimation process:

1. the initial model λ defines a critical point in likelihood function (null gradient), and in this case $\hat{\lambda} = \lambda$

2. $Prob(O|\hat{\lambda}) > Prob(O|\lambda)$ and hence we have found a new model for which the observation sequence is more likely.

The re-estimation theory does not guarantee that in the first case the critical point is also an absolute maximum of the likelihood function. Furthermore, the likelihood function is often a very complex function of many parameters that make up the model λ, and hence finding a local maximum in the re-estimation process is easier than finding an absolute maximum.

5.3.3 HMM Assumptions

The HMM theory is based on some a priori assumptions about the structure of the speech signal [3].

- The speech signal is supposed to be approximately represented by a sequence of observation vectors in time.

In our case, as described in section 4.2, these vectors are the cepstral coefficients evaluated on consecutive overlapping frames of speech of the order of 10ms long. As already pointed out, the speech signal can approximately be considered to be stationary on this time scale, hence this is a good assumption.

- The speech signal can be represented by a finite number of mixture Gaussian probability distributions

This is a rough approximation, in fact the speech signal varies considerably, especially in a speaker independent context.

- The observation vectors are independent.

This is probably the most serious approximation in the theory. The observation vectors do are correlated for many reasons: consecutive vectors refer to the same phoneme, to the same prosodic feature, to the same speaker, etc. Results can be improved by including dynamic features in the speech parameterization, as described in Section 4.2, because they hold information about the local signal context.

- The probability of the transition from the current distribution to the next distribution does not depend on any previous or subsequent distributions (first order HMM).

The last assumption is also untrue.

In spite of these rough approximations, HMM performance is often very high in speech applications.

5.3.4 HMM Applications

HMM-based recognition systems normally make use of multiple HMMs. Models are usually small (up to four or five states), and their structure is usually left-to-right (non ergodic). These models are used to model short units of speech, usually phones or words. Word based HMMs are used in small vocabulary applications, such as digit recognition, while sub-word HMMs must be employed when the number of words to be considered is too high. Sub-word systems are also more flexible, because it's always possible to add words in the dictionary on condition that a pronunciation (list of phones) for those words is provided.

5.3.5 State Output Probability Distribution

The state output probability function (B in Section 5.2) used in most systems is based on Gaussian distributions. Often many weighted Gaussian distributions are employed in the attempt to represent the high speech signal variability. Considering only one element in the output frame vector (in our case this is one of the mel-cepstral coefficients), we have

$$b_j(k) = \sum_{h=1}^{H} c_{jh} G(m_{jh}, \sigma_{jh}^2)$$

where $G(m_{jh}, \sigma_{jh}^2)$ is the Normal (Gaussian) distribution of mean m_{jh} and variance σ_{jh}^2 and H is the number of terms in the mixture distribution. In Figure 5.2 an example of this kind of probability distributions is depicted. The speech signal variability depends on many factors. The first source of variability is the dependence of the phoneme pronunciation on the context. Many other sources of variability can be mentioned regarding, for example, speaker characteristics (sex, age, region). The introduction of many Gaussian terms allows for higher resolution, but slows down

the recognition process, making real time applications difficult to be run even on fast machines. The second limit to this method is the need of a large database to obtain a robust estimation of model parameters.
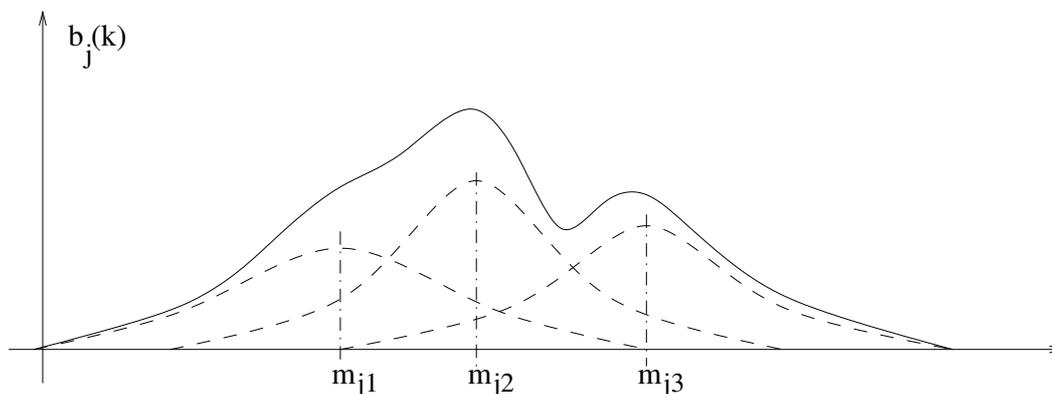
Figure 5.2. Multi mixture Gaussian distribution

5.3.6 Context Dependent HMMs

In simple phoneme-based systems, only one model per linguistic phoneme is used. However, phonemes differ considerably depending on their context, and such a simple modeling is too poor to hold this variability. A possible solution is to add Gaussian distributions, as already discussed. Each distribution can model a different phoneme realization. The problem of this method is that in each model it is impossible to know which Gaussian distributions correspond to a particular context and hence in computing the data likelihood given the model, all parameters must be considered and the recognition process is slowed down.

Another solution is to create a different model for each phoneme affected by a particular context. A different label is associated with each context dependent model. During recognition only one of these models would be used depending on the actual context, and, since each context dependent model require a lower number of parameters[4], the process is sped up. Two kinds of context expansion have been used in this work depending on whether cross word contexts are taken into account or not. To make this distinction clearer in Table 5.1 samples of label files, built according two different methods, are showed. The source sentence is: `...omedelbart att granska fallet.` The two methods are compared in the table to the context independent expansion. Note how in the third case the last phone in the previous word, or the first phone of the next word are used to complete the triphone context of models at word boundaries. Noise models are never included

---

[4]Because it doesn't have to hold information about the context variability

Context independent	Within word CE	Cross Word CE
...	...	...
A	A+T	T–A+T
T	A–T	A–T+G
#	#	#
G	G+R	T–G+R
R	G–R+A	G–R+A
A	R–A+N	R–A+N
N	A–N+S	A–N+S
S	N–S+K	N–S+K
K	S–K+A	S–K+A
A	K–A	K–A+F
#	#	#
F	F+A	A–F+A
A	F–A+L	F–A+L
L	A–L+EO	A–L+EO
EO	L–EO+T	L–EO+T
T	EO–T	EO–T
#	#	#
sil	sil	sil
.	.	.

Table 5.1. Different phone-level expansion methods (CE=Context Expansion)

in the context of any phone model. This is because, in our opinion, phoneme pronunciation depends more strongly on the last or next phoneme uttered then on involuntary speaker noise, such as lip smack. In the case of environmental noise there is no reason to believe that this noise can influence the pronunciation. In the following we refer to these models as "context free" models.

5.3.7 The Data Scarcity Problem

Context information can be taken in account in different ways when building context dependent models, including phonetic context, word boundary information, or prosodic information. The number of models rises very rapidly with the amount of context information included. If both right and left context are included, for example, and if we consider 46 phonemes in Swedish, the number of models to be included in the model set may be up to 97336[5]. Many of these contexts may not occur even in large databases, or there may be not sufficient occurrences to allow a robust estimation of the corresponding model parameters. They may also never be needed.

There are three common methods to solve these problems:

Backing off Context dependent models are estimated only where sufficient training data exists. When unavailable context dependent models are required, reduced context or monophone models, for which sufficient training data was available, are used instead. The problem with this method is that the resulting accuracy is lower when using monophone models.

---

[5]If we consider 42 phonemes the maximum number of models is 74088.

Top-Down Clustering A priori information about phonetics is used to tie parameters of models considered to be acoustically similar. The method is based on a decision tree which can be used also to generate unseen models. Clustering stopping criteria can be used to ensure that each tied model can be robustly estimated.

Bottom-Up Clustering The clustering process is based on the parameter similarity in each model. Similar models are tied to form a single model. Even though this method makes a better use of the training data, it does not provide models for contexts not seen in the training material.

The top down clustering method, also called *tree clustering* method, has been employed in this work and it is described in the following section.

5.3.8 Tree Clustering

A phonetic decision tree [18, 3, 17, 1, 13, 6] is a binary tree containing a yes/no question for each node. For each base phone `ph` the pool of states belonging to models in the form `lc-ph+rc` is successively split according to the answer to each question, and this continues until the states have reached the leaf-nodes. States which are in the same leaf in the end of this process, are tied together. Questions are in the form: "is the left/right context (`lc/rc`) in a set P?"[6].

The question tree is built on the run: first a list of all possible questions is loaded. Then for each set of triphones derived from the same base phone, corresponding states are placed in the root. A question (in the question list) is chosen and two leaves are created from the root for each answer to this question. The criterion for choosing the question is intended to (locally) maximise the increase of likelihood $L(D|S)$ of the training data D, given the set of tied distributions S in the just created leaves. In practise, since the algorithm is restricted to the case of single Gaussian distributions, $L(D|S)$ can be evaluated from the means, variances and state occupation counts, without reference to the training data. This in the assumption that the assignment of observation vectors to states is not altered during clustering, and thus the transition probabilities can be ignored. The process repeats for each new node until the increase in log likelihood falls below a specified threshold, say `tbth`. High values of `tbth` correspond to a short tree, and hence to a small number of states in the model set. In Figure 5.3 a graphical representation of the question tree for the phone `J` is shown. The threshold in this case had an high value (1100) so that the tree has only two levels. Only a few states are generated and many states in different models share their parameters. More realistic values for the threshold bring to a deeper ramification, and to more accurate models.

---

[6]In appendix D the definition of the sets P for phonetic classification used in this work is showed

left context ——— *−J+* ——— right context

Q:L_VOICED

Y N

Q:R_VOWEL Q:R_BACK

Y N Y N
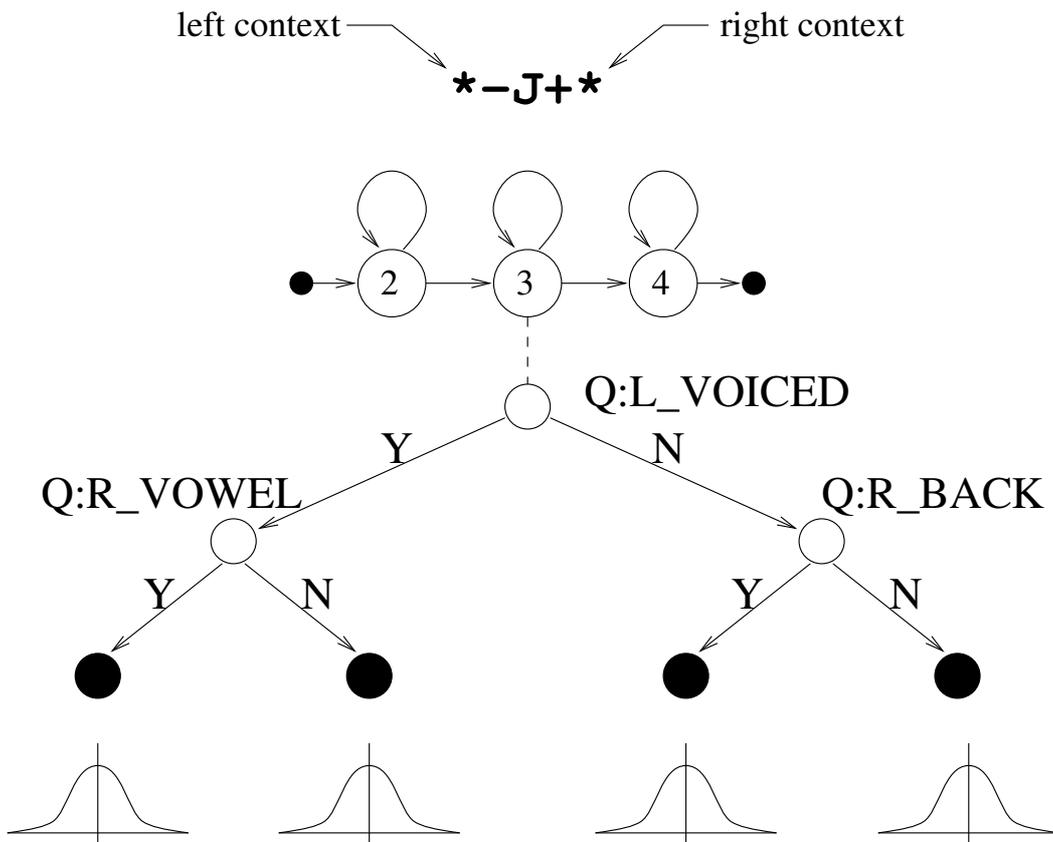
Figure 5.3. An illustration of a decision tree.

Chapter 6

Model Set Building

In the attempt to increase model resolution several experiments have been performed investigating different possibilities of improvement. When building statistical models, the possibility of high performance is always based on a trade off between model complexity and the amount of avaliable data for parameter estimation. A good choice for model structure, based on the comprehension of the sources of variability in the physical phenomena, can help increasing the ability of the model to fit the problem of interest with the minimum number of parameters.

During this work two main directions of improvement have been followed. Starting from simple context independent models the first method is to add Gaussian terms in the state output probability distribution. In this case the re-estimation procedure is completely responsible for adapting model parameters to different sources of variability in the speech signal. As a second method context dependent models have been tested. This method is based on the assumption that one of the most important sources of variability in each phoneme realization is its context. If this assumption is true the re-estimation procedure deals only with speaker related variability, which can be modeled in an easier way. The two methods are also employed in combination.

Furthermore each experiment has been worked out using two phonetic classification methods called "old" and "new" lexicons for reasons related to the evolution of the SpeechDat project [4]. The old lexicon avoids the use of "retroflex allophones" (except for **2S**) described in Section 2 because of their low occurrence frequency in the database, while the new lexicon introduce them again making for to two different model sets for each experiment.

To make the distinction between models obtained with different methods clearer, each experiment is marked with a label in the form `[c][n]exp[a]`, where `n` stands for "new SpeechDat lexicon", `c` means that models are clustered to reduce the number of parameters and `a` means that all possible triphones are added to allow network expansion.

The first model set including context independent models is marked as `mb` (monophones). Context dependent models are obtained from monophones by duplicating

each phone model as many time as the number of different contexts occurring in the training material. These model sets are marked with a `tb` label (`t`riphones word `b`oundaries) if only within word contexts are considered. `ntb` is the mark for the same experiment with the new lexicon.

If cross word context information is included in the model set, a `tnb` mark (`t`riphones `n`o word `b`oundaries) is used (`ntnb` for new lexicon models).

After monophones are duplicated for all possible contexts, the next step is to apply the clustering procedure to reduce the number of parameters. This procedure results in the creation of a new models set in which many states are clustered, i.e. many models share the same parameters for corresponding states. Marks for model sets obtained with this method can be derived from the triphons marks adding a `c` (`ctb`, `cntb`, `ctnb`, `cntnb`).

Finally all possible context dependent models are synthesized according to the tree clustering procedure and added to each context dependent model set to allow context expansion during recognition. Marks in this case are `ctba`, `cntba`, `ctnba`, `cntnba`.

In this chapter the steps needed to perform these experiments are described.

6.1 Data Preparation

6.1.1 Speech Files

SpeechDat audio files (8kHz, 8bit/sample, A-law) are coded according to the procedure dicussed in Section 4.2 obtaining vectors of 39 components (12 mel-cepstral coefficients + 1 energy + 13 delta coefficients + 13 acceleration coefficients) for each speech frame.

6.1.2 Label Files

SpeechDat transcriptions are used to generate a word level label file including transcriptions for each audio file. This label file is then used to create phone level transcriptions. Word labels are first expanded into phone labels according to the pronunciation specified in the lexicon file. Two separate phonetic transcriptions are obtained with the old and new lexicon and stored in separate files. Then context dependent phonetic transcriptions are created expanding each phone label depending on the context. Four label files are created for old/new lexicon and for within/cross-word context expansion. Samples of these files are shown in Table 5.1, page 31.

6.2 HMM Setup

Two kinds of segments must be considered when building acoustic models: *target* speech regards segments of the audio signal which contain useful information about the message we want to transcribe, and *non-target* speech regards segments of the audio signal containing noise generated by various sources and which do not add

information to the message. In our system *target* speech is modeled at the acoustic level as a sequence of phones. *Non-target* speech is modeled by six dedicated HMMs. Four HMMs model SpeechDat kind noise (Chapter 3), while the other two are for silence segments and word boundaries.

The standard structure is left-to-right hidden Markov models with a maximum of four emitting states. Since each frame in the data stream consists of a 39 elements vector (Section 6.1), each state has 39 associated probability distributions, i.e. every parameter[1]

$$b_j(k) = Prob(O_t = v_k | X_t = s_j)$$

is a 39 dimensional probability distribution in which every component, in our case, is supposed to be statistically independent from the others.

Every component in this multi dimensional distribution can be either a simple Gaussian or a weighted sum of N Gaussian distributions (multiple mixture).
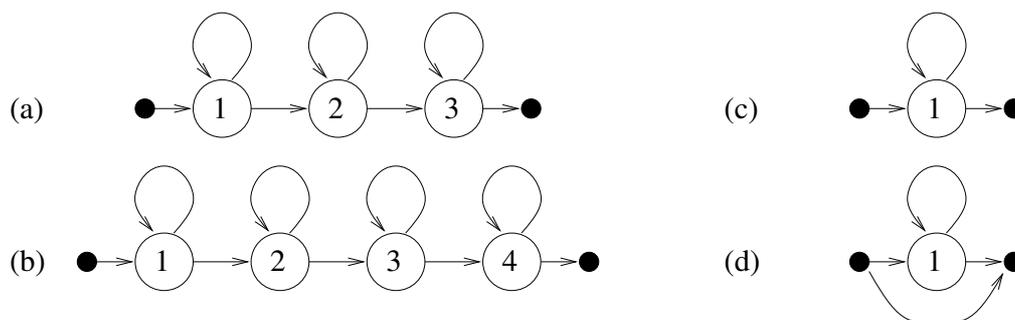
Figure 6.1. Model structure

6.2.1 Target Speech

All phones, except plosives, are modeled by a three emitting state HMM as depicted in Figure 6.1(a). The choice of topology in HMM applications is often made in the attempt to obtain a good balance between forward transitions and transitions back to the same state. This balance seems to be more important than the correspondence between different states and different parts of the same phoneme realisation. This is true for all the steady sounds, such as vowels, fricatives, nasals and so on. The sound produced in the case of plosives (**B**, **D**, **2D**, **G**, **K**, **P**, **T**, **2T**), on the other hand, has an important temporal structure, and states in the corresponding HMM should match each different acoustic segment. For this reason each plosive is modeled by an HMM with four emitting state (Figure 6.1(b)).

---

[1]See section 5.2

6.2.2   Non-Target Speech

Garbage

As already described in section 3.1.2, SpeechDat adopts the convention to mark
with a "*" mispronounced words, with "**" unintelligible words and with a "~"
truncated words. For all these occurrences there is no entry in the lexicon, i.e. no
pronunciation is provided. To allow the use of files containing these symbols during
training, a **garbage** model has been created. It has only one state with a mixture
of many Gaussian distributions, because it is supposed to match a wide range of
sounds. As we discuss later this model is trained on generic speech, and it should
win the competition with models of specific sounds. The **garbage** model is not
used in recognition because only "clean" speech files are used for this task.

Extral, Noise, Öh

Noise models are: **extral** (SpeechDat mark: `[spk]`) for speaker noise such as lips
smack, **Öh** (`[fil]`) for hesitation between a word and another, **noise** (`[int]`)
for non speaker intermittent noise typical of the telephone lines. There is not a
particular model for stationary noise (`[sta]`) because this disturbance is extended
to the whole utterance. However, when using noisy files in the training process, the
stationary noise characteristics take part in the model parameter estimation, and
are hence held by those parameters.

Sil

The **sil** model is a three state HMM (Figure 6.1 (a)) trained on silence frames of
the utterance. In the recognition task it is used at the beginning or at the end of a
sentence in the attempt to model the extra time in the recording session over the
spoken utterance.

#

This model is employed for word boundaries: it has a symbolic use, representing
the boundaries between words at the phone level and allowing different context
expansion methods as already described. A second reason for its use is to model the
silence between one word and another. In continuous speech these silence segments
can be very short. For this reason # has only one emitting state as depicted in
Figure 6.1(d). This state is tied (shared parameters) with the central state of the
sil model. Furthermore a direct transition from the first (non emitting) state to the
last (non emitting) state is allowed in the case that words are connected without
any pause.

6.3   Training The Models

Training consists of applying an embedded version of the Baum-Welch algorithm discussed in Section 5.3.2. For every speech file (output sequence) a label file with a phoneme transcription is loaded and used to create an HMM for the whole utterance concatenating all models corresponding to the sequence of labels. This is possible thanks to a property of HMMs: researchers have discovered that if an observation sequence is presented to a strings of HMMs they will tend to "soak up" the part of the observation sequence corresponding to their word or phone [9]. This is true when the model set parameters are not too far from the optimal values. When initializing the model set, a different method must be employed. This was not necessary in this work because the first model set has been obtained modifying by hand an already trained set of models [12], and hence it is still able to perform a crude alignment. The only completely new model is **garbage**. In this case the model has been trained separately on generic speech files, and then included in the model set.
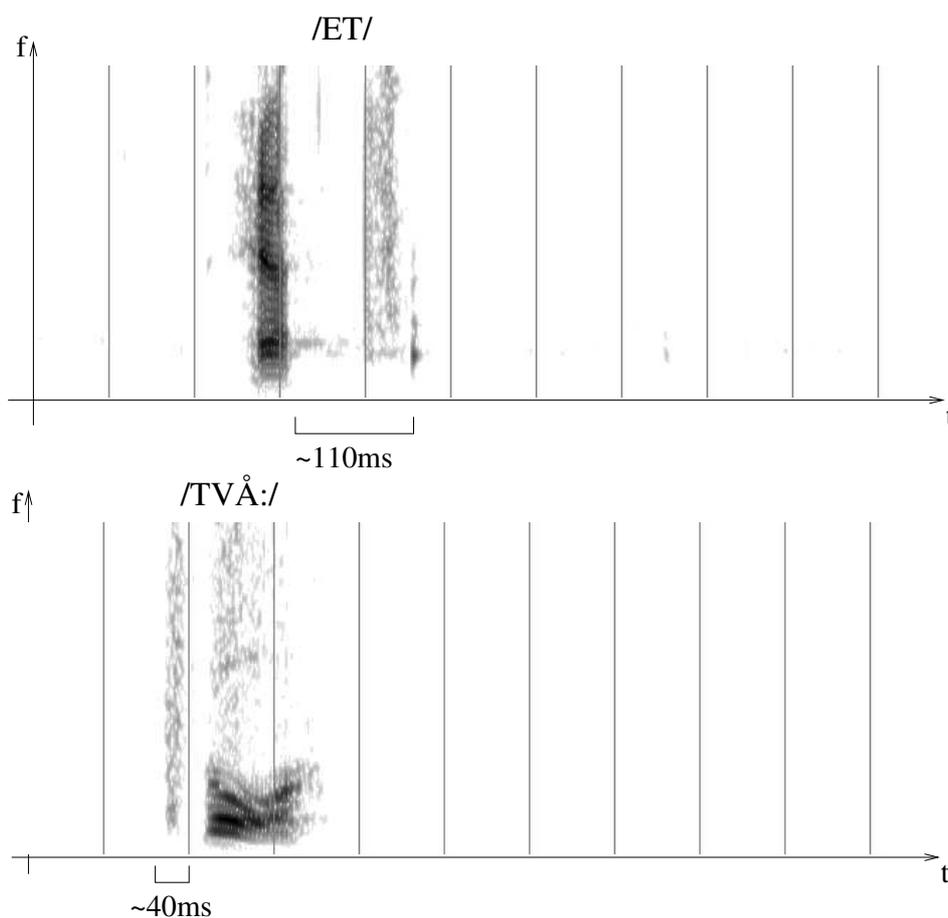
Figure 6.2. Spectrogram of the words "ett" (one) and "två" (two)

The model set we modified [12] contains the concatenation of two HMMs with three emitting state for each stop consonant. The first model for each stop (named with an uppercase letter: B, D, G, K, P, T) models the occlusion segment in the stop phone, while the second (lowercase letter: b, d, g, k, p, t) models the release segment. In our opinion this method can be inaccurate because of the fast time evolution of plosives. Figure 6.2 shows the spectrogram of the words "ett" (one) and "två" (two). A rough measure of the /t/ stop duration is given in the figure. If we consider 10ms frame interval and six states in the T t model, the minimum duration for model evolution is 60ms, if no transition back to the same state occurs. As we see in the figure this model can be good for the realization of the T sound in the "ett" word, but it is to "slow" for the realization of the same phone in the "två" word. This is one reason why each couple of stop models (B b for example) has been substituted with a four emitting state model. A second (practical) reason is that it is easier to represent stop consonants with one symbol in the dictionary than with two. In the attempt to preserve the ability of the model set to align the training data, allowing the standard re-estimation procedure, stop models are not created ex-novo. They are derived from the old model set in the following way. For each four state model the first emitting state is obtained copying the last state in the corresponding occlusive model. The other three states are copied from the corresponding release model. The reason for this choice is that, even though the occlusive segment is often longer in time than the release segment, the first one consists of an approximately stationary sound, while the latter include most of the time varying evolution (explosion).

Different model sets obtained by successive re-estimation procedures are stored in different directories to allow a comparative evaluation of their performance and to figure out when to stop the iterative procedure.

Before proceeding the description of model set creation, the evaluation method must be introduced.

6.3.1 Development Tests

During training development tests are an important tool in the attempt to compare different experiments and to suggest new possible ways of improvement. These experiments consist of a word level recognition on a small subset of speakers reserved for this task. The number of speakers (50) involved in these tests is too low to guarantee a statistical consistency of the results, nevertheless these tests can be used for a comparative evaluation of different model sets[2]. When scoring the system, two alternative parameters are taken into account: correct words and accuracy. Their definition depends on the the algorithm used to align the output of the system to the reference transcription. This algorithm is an optimal string match based on dynamic programming [18]. Once the optimal alignment has been found, the number of substitution errors (S), deletion errors (D) and insertion errors (I) can

---

[2]See Section 3.1.4 for data on speaker subsets.

be calculated. Correct words (PC) are then defined as:

$$PC = \frac{N - D - S}{N} \times 100\%$$

While the definition for accuracy (A) is:

$$A = \frac{N - D - S - I}{N} \times 100\%$$

Files chosen for evaluation contains (see chapter 3) isolated digits, connected digits and natural numbers. A list of all words in the development test files is included in appendix C.

6.3.2 Monophones

Nine iterations of the Baum-Welch re-estimation have been performed. As showed in Figure 6.3 recognition performance in terms of word accuracy is improved until the fifth iteration in the case of old lexicon models (`mb`). Further iterations can be avoided since they don't bring better results. In some cases, as we will see, the performance is even reduced. This is because, with too many iterations, HMM parameters tend to fit too well the training data (and hence the training speaker characteristics) and have no more freedom to generalize to new speakers (evaluation data). In terms of Gaussian parameters it means that the variances tend to be too low (narrow Gaussian shape) to include new speaker variations. The reason why
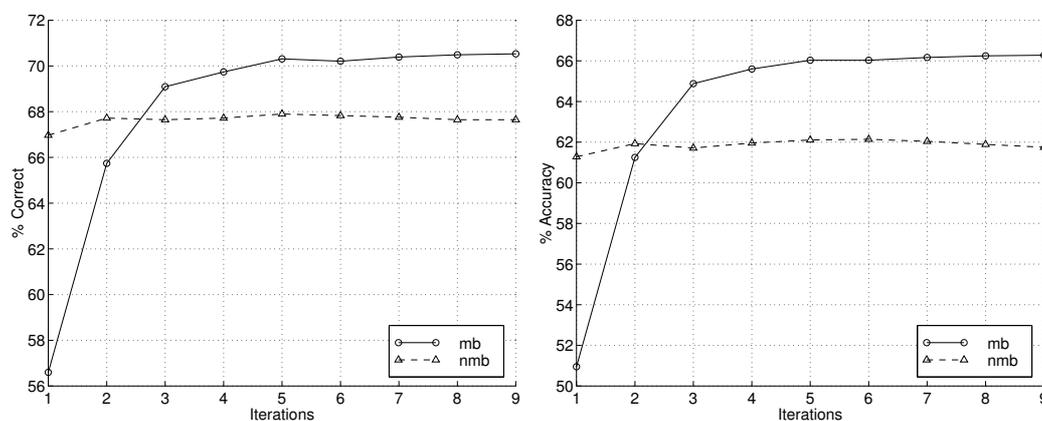
Figure 6.3. Development tests, monophones: correct words (left) and word accuracy (right) after each of nine iterations

the accuracy curve is flat for the new lexicon experiment (dashed line) is that this set of models has been obtained from the third iteration on the old lexicon model set copying each model (**T**, **D**, **N**, **L**) twice and naming the resulting models **T**, **2T**, **D**, **2D**, and so on. For this reason the new model set in the first iteration has

been already subjected to four iterations (three with the old models and one with the new) of the Baum-Welsh reestimation.

The first attempt to increasing accuracy was to add Gaussian mixture terms. Results obtained with this method are shown in Figure 6.4 where models with 1, 2, 4 and 8 Gaussian terms are compared. Adding Gaussian distributions consists of splitting state output distribution of the previous model set (ninth iteration). The splitting procedure consists of selecting distributions with the higher associated weight and copying them twice in the new model set. The weight is divided by 2 for each new distribution, while the mean is perturbed. Analytically:

$$cG(m, \sigma^2) \rightarrow \frac{c}{2}G(m + 0.2\sigma, \sigma^2), \frac{c}{2}G(m - 0.2\sigma, \sigma^2)$$

The process is repeated until the wanted number of mixture terms is reached. In these figures it is possible to see how models including the four allophones **2T**, **2D**, **2N**, **2L** (dashed line) have a worse performance if compared to the old lexicon models. This difference tends to decrease if the number of mixtures terms is increased.
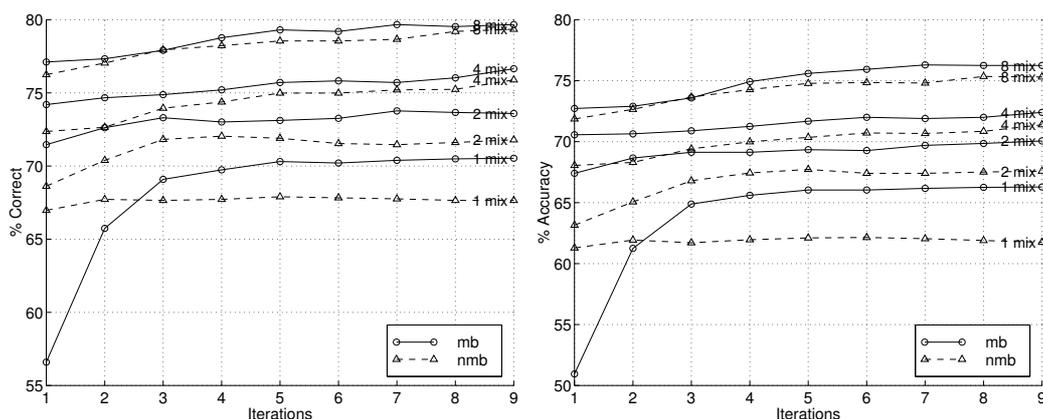
Figure 6.4. Development tests, monophones: correct words (left) and accuracy (right) for 1,2,4,8 mixture terms

6.3.3   Triphones

The construction of context dependent models has been shown to be a good alternative method in the attempt to improving accuracy: Two expansion methods have been tested:

- within-word context expansion

in which phonemes at word boundaries are expanded as diphones (see Table 5.1 page 31). This method is easier to apply in the recognition phase because models

created during network expansion depend only on the words in the dictionary and not on their sequence in the sentence hypothesis. This means that avoiding unseen context dependent models is easy, especially if the words included in the recognition task are also present in the training data.

- cross word context expansion

This method results in the generation of a lower number of diphones (only phonemes at sentence boundaries are expanded as diphones). On the other hand, the number of triphone occurrences is increased, rising the context information, and sometimes the number of occurrence for a single model. In Table 6.1 the number of triphones and diphones for each experiment is presented. From the context expanded master

Experiment	triphones	diphones	tot	theoretical
tb	8795	1029	9835	74088
tnb	15618	1150	16775	
ntb	9176	1063	10250	97336
ntnb	16559	1187	17753	

Table 6.1. Number of triphones and diphones in different model sets

label file (see Table 5.1) a list of all triphones and diphones included in the training files is created. Then for each label in the list a copy of the corresponding monophone model from the old model set is included in the new model set. For example in case of a label in the form `lc-ph+rc` (triphone) the model `ph` (monophone) is copied and named "`lc-ph+rc`" for each occurrence of different `lc` and `rc`. In the end we obtain a new model set in which all triphone models corresponding to the same phoneme are identical. Context free models are simply copied as they are in the old model set. The model set obtained has been subjected to a first iteration of the Baum-Welsh algorithm. For most models the training data was not sufficient, thus a tree clustering procedure had to be applied. Several threshold values have been used in order to find a good trade-off between the number of states in the model set (model variability) and size of available data (estimation robustness). In Figure 6.5 the number of states generated in the clustering process is related to the threshold values used. Model sets obtained with different threshold values have been trained separately. Then these models have been tested to find the optimal value for the threshold. Results are showed in Figure 6.6. The optimal model set contains 2020 states in the case of within-word context expansion and the old lexicon (tb) and 2440 states for the new lexicon (ntb); and 1758 and 2862 respectively for old and new lexicon (tnb and ntnb) and cross-word context expansion. Figure 6.6 also shows that models not including retroflex allophones perform better also in the case of context dependent modelling.

The best models have been developed by adding Gaussian mixture terms to output probability distributions. Results obtained with this method are showed in
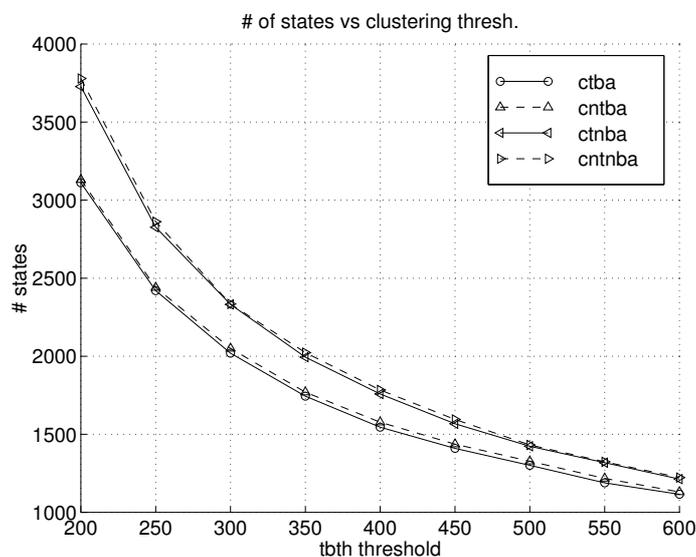
Figure 6.5. Number of states versus tree clustering threshold
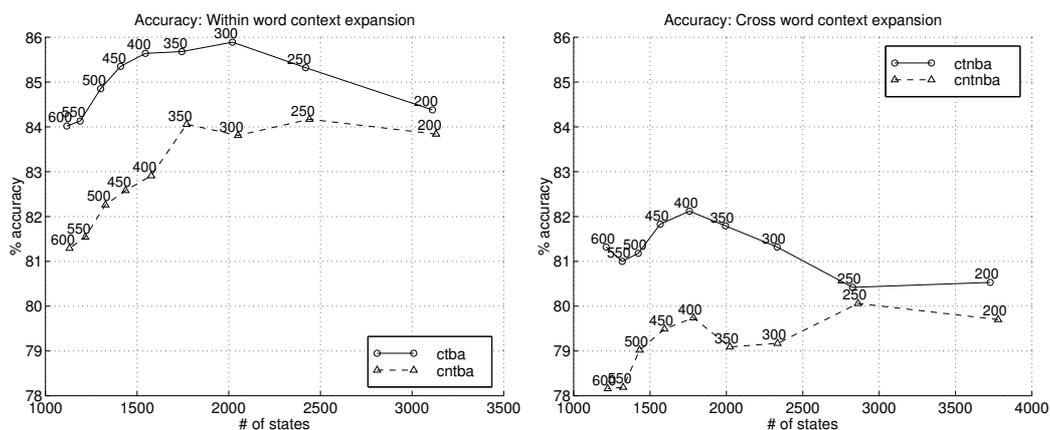
Figure 6.6. Clustering threshold optimization

Figures 6.7 and 6.8 respectively for within-word and cross-word context expansion. Figures show nine iterations and models with 2, 4 and 8 Gaussian distributions per mixture. As in the case of monophones, the difference in performance between models including or excluding retroflex allophones is reduced as the mixture size is increased. Within-word models perform better than cross-word models, probably because the number of contexts is lower (6770 models instead of 9681), allowing a more robust parameter estimation. Anyway these results are affected by the specificity of the task. The advantage of using cross word context expansion would

probably be higher in a generic speech recognition task in which an higher number of words is involved and sentences are uttered in a more continuous way.
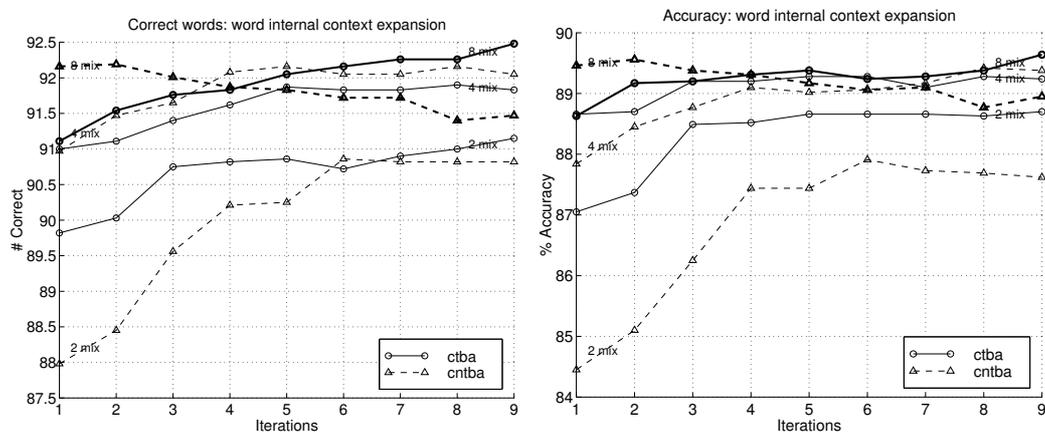
Figure 6.7. Development tests, triphones within-word context expansion: correct words (left) and accuracy (right)
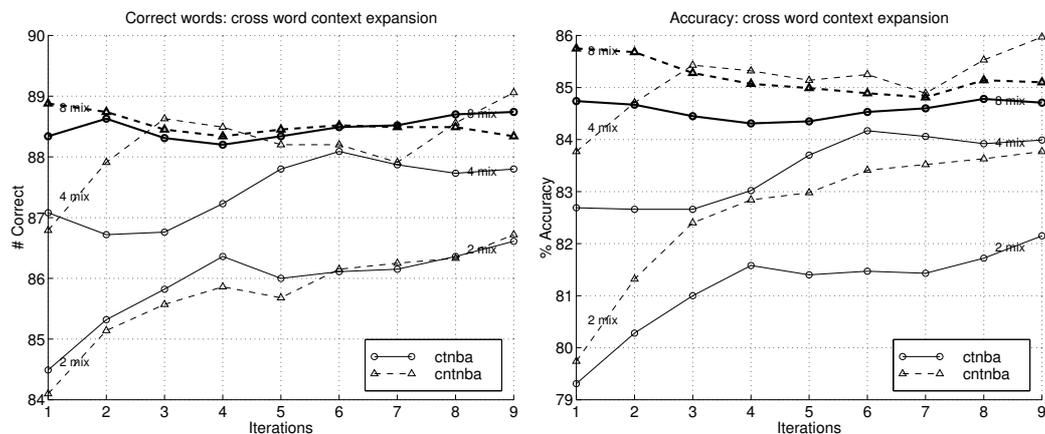
Figure 6.8. Development tests, triphones cross-word context expansion: correct words (left) and accuracy (right)

Chapter 7

Results, Analysis and Discussion

This chapter begins with a definition of the "difficulty" of the problem of speech recognition (Section 7.1). Then results are described and analyzed first for the full evaluation subset of speakers in Section 7.2 and then with regard on the individual speaker characteristics in Section 7.3. Since this work has been the first attempt to train acoustic models on the Swedish SpeechDat database, the only way to compare results with previous works is to refer to different experiments. In Section 7.4.1 results are compared to those obtained with a database built for a different purpose (the Waxholm database) but for the same language (Swedish). In Section 7.4.2, instead, the comparison is made for a similar database (the Norwegian SpeechDat database), but for a different language (Norwegian).

7.1   Language Model Perplexity

When evaluating the performance of a speech recognizer, quantifying the difficulty of the recognition task is an important aspect to consider. One measure can be obtained from the so called *perplexity* of the language model[9]. Perplexity is an indirect measure of the entropy of the language: the higher the entropy, the more information must be extracted from the recognition process and the more difficult the task is. In other words, when employing language constraints such as a grammar, the uncertainty of the content of sentences is reduced, and the recognition task is made easier. Perplexity as defined in [9] involves the estimation of the probability that a particular sequence of words is generated by the language source. If x is a symbol put out by a source with probability $P(x)$, then the entropy of the source is defined as

$$H = -[\sum_{x=1}^{L} P(x) \log P(x)]$$

where L is the number of possible symbols. If we consider sequences of symbols the entropy is defined as

$$H = -\lim_{n \to \infty} \frac{1}{n} [\sum P(x_1, ..., x_n) \log P(x_1, ..., x_n)]$$

where the sum is over all sequences $x_1, ..., x_n$. If the source is ergodic, one sufficiently long sequence of symbols can be used to deduce its statistical structure, hence

$$H = -\lim_{n \to \infty} \frac{1}{n}[\log P(x_1, ..., x_n)]$$

Finally *perplexity* is defined as:

$$PP = 2^H = P(x_1, ..., x_n)^{\frac{-1}{n}}$$

where the limit has been removed for real applications. In our case the sequence $x_1, ..., x_n$ correspond to the evaluation corpus (concatenation of sentences in the evaluation files) while $P(x_1, ..., x_n)$ can be evaluated from the word bigram probability, estimated on the training material. Perplexity for our case is around 40, classifying the task as a medium perplexity recognition problem [19, 7]. Normally digit recognition problems have in English $PP = 11$. In our case other words are included allowing for example natural number recognition, and the same grammar constraints are used for all tasks.

7.2 Overall Results

From the development tests (Chapter 6) the best models were selected and tested on the evaluation material. Results obtained with these tests are showed in Table 7.1. In the table correct words and accuracy are reported for each experiment. The best

experiment	1 mix		2 mix		4 mix		8 mix	
	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc
mb	69.4	66.4	72.6	69.5	75.6	72.3	78.9	76.0
nmb	68.1	63.1	71.5	67.9	75.1	71.3	79.1	75.5
ctba			89.5	87.4	90.7	88.5	90.8	88.6
cntba			89.1	86.4	90.3	88.1	90.5	88.3
ctnba			86.1	81.8	87.8	84.0	88.4	84.8
cntnba			86.8	84.2	88.4	86.1	88.9	86.5

Table 7.1. Overall results: evaluation data correct words (Corr) and accuracy (Acc) for monophones, old lexicon (mb), monophones, new lexicon (nmb), triphones, old lexicon, within word CE (ctba), triphones, new lexicon, within word CE (cntba), triphones, old lexicon, cross word CE (ctnba), triphones, new lexicon, cross word CE (cntnba).

result (88.6% of accuracy) is obtained with within-word context expanded models and eight mixture terms. As can be seen in the table, monophone accuracy rises when the number of mixture terms is increased from four to eight. This means that probably better results can be obtained if the number of mixture terms is further increased. In the case of context dependent models the increase of accuracy from

four terms models to eight terms models is quite low. Old lexicon models perform better in general than new lexicon models. Models excluding cross-word context information perform better than models including it.

Since the number of development speakers is low, plots of accuracy and correct words as functions of the number of iterations are also reported to check if performance follows the same trend on a larger and independent data set. Figure 7.1 shows results for monophone models. In each plot models based on the old lexicon seem to perform better than models including the retroflex allophones, if the number of Gaussian distributions is the same. Only in a few cases more than five or six iterations are useful to improve performance. Figure 7.2 shows results for within
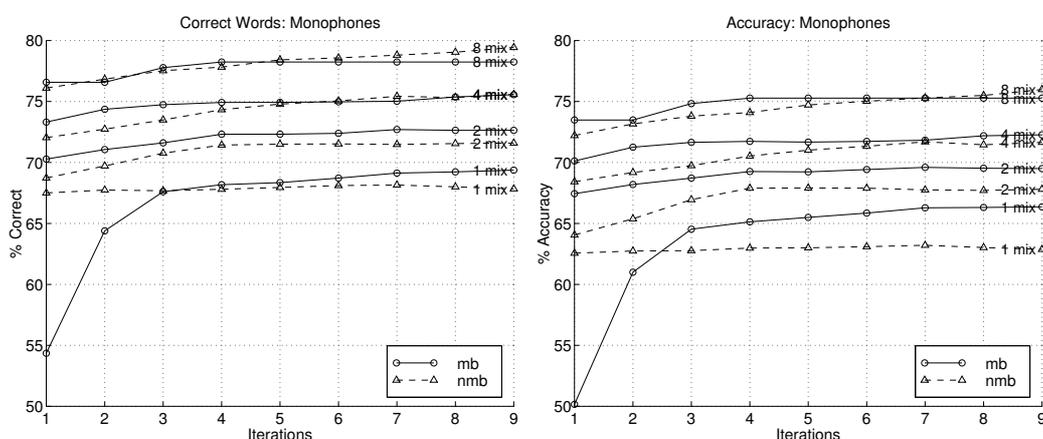
Figure 7.1. Evaluation test, monophones: correct words (left) and Accuracy (right).

word context expanded triphones. In this case curves are not as regular as in the monophone case. This may be because the amount of training data for each model is not as consistence as in the monophone case. Many times, the best result is obtained for a different number of iterations if compared to the development tests results (Figure 6.7 page 45). For example, in the case of old lexicon, eight-term mixtures (solid thick line) the best result is obtained after five iterations. Then accuracy falls down, while in the development test it continues rising till the ninth iteration. This shows how these models, depending on the lower amount of data they are trained on, could be (lightly) dependent on the speaker characteristics and also shows that we should be careful when considering results on the fifty speakers development subset. Once again the old lexicon models perform better, but this difference is slightly reduced when adding Gaussian distributions. Figure 7.3 shows results for cross word context expanded models. In this case a comparison between old and new lexicon models is more difficult because the number of states contained in each model set is ... different (1758 states for the old lexicon: `ctnba` and 2862 states for the new lexicon: `cntnba`). This difference has been caused by the non regular shape of the accuracy curve in the tree clustering optimization procedure
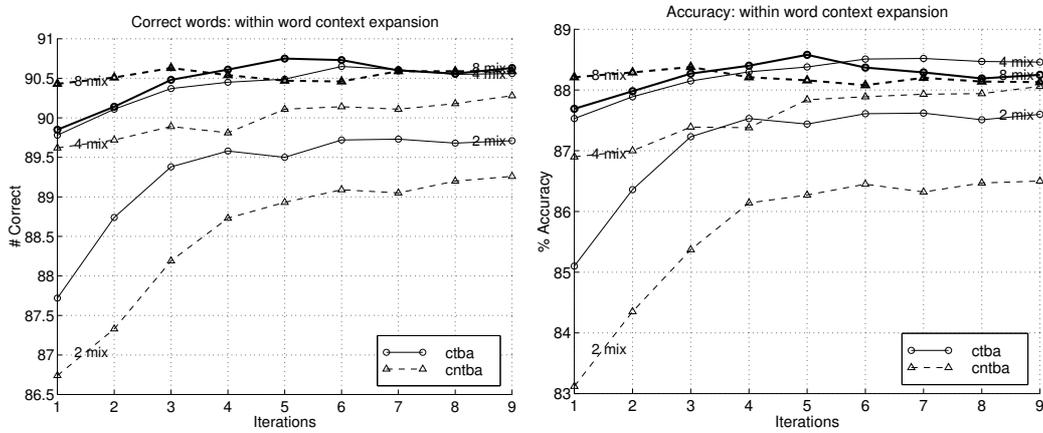
Figure 7.2. Evaluation test, triphones, within word expansion: correct words (left) and Accuracy (right) for 2, 4, 8 mixture terms.

(Figure 6.6, page 44). In the case of models including the retroflex allophones (new lexicon: `cntnba`), the accuracy curve in Figure 6.6 has two maxima for 1784 and 2862 states in the model set. The latter model set has been chosen for further improvements because the accuracy is higher. Referring to Figure 7.3 unexpected results are obtained: the `cntnba` model set (2862 states, dashed line) performs better than the `ctnba` model set (solid line, 1758 states). This difference in results seems to be more related to the difference in number of states than on the different lexicon.
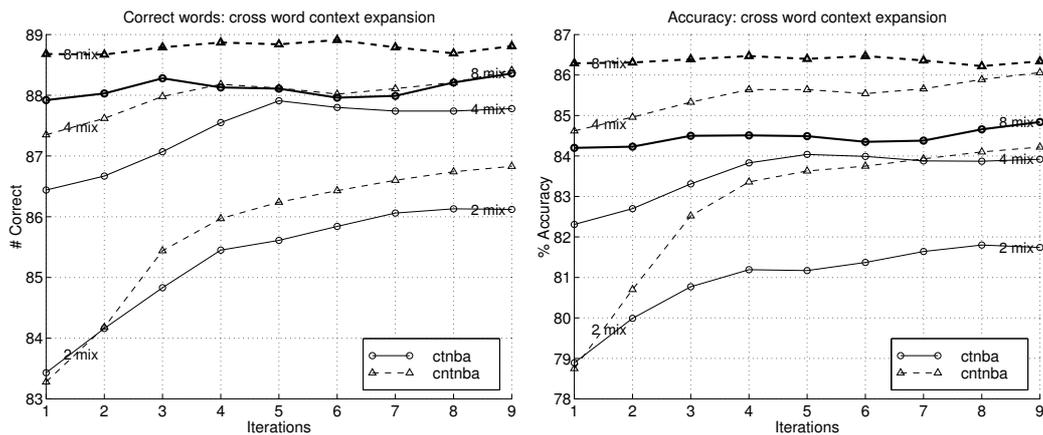
Figure 7.3. Evaluation test, triphones, cross word expansion: correct words (left) and Accuracy (right) for 2, 4, 8 mixture terms.

7.3 Results per Individual Speaker

SpeechDat database is built on a wide range of speaker characteristics (Chapter 3). For this reason it is interesting to show per speaker results. Often in speaker independent recognition tasks, speakers are divided into "goats" and "sheeps" depending on results obtained. "Goats" are those speakers for which bad results are obtained, while "sheep" speakers are well recognized by the system. The definition of the threshold separating these groups is arbitrary and depends on the application. In
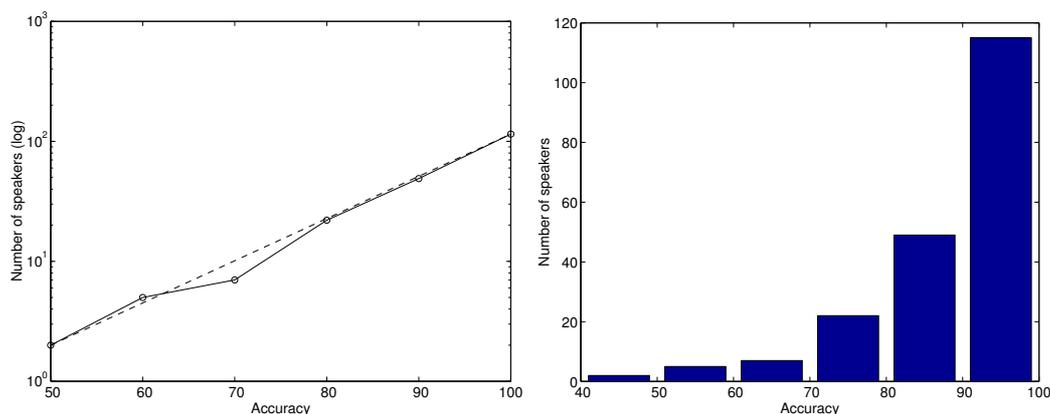
Figure 7.4. Number of Speakers per accuracy.

Figure 7.4 (right plot) the number of speakers for which results are in ranges of ten percent of accuracy are shown. No speaker in the evaluation subset has results below 40% of accuracy. If we set the boundary between "goats" and "sheeps" at 80% of accuracy, 36 speakers of the 200 in the evaluation subset belong to the "goats" group while the other 153 are "sheeps". In the left plot of the figure the same data is plotted in a logarithmic scale showing a linear behavior ($y \approx m + ax$). Knowing m and a can be useful to predict results when new speakers are added to the evaluation set, or to evaluate new developments in the systems. Figure 7.5 reports results according to sex and age of the speaker, the left plot include results on female speakers and the right plot on male speakers. In the figure the left bar for each age group indicates the accuracy, while the right bar indicates the percentage number of correct words. An error bar is also depicted showing the standard deviation in each group (the number of speakers in each group is reported in Table 3.2 page 14). Results seem to be sex independent, even though female speakers are better recognized (probably because they are more numerous in the database). Figure 7.6 shows results depending on the region speakers call from. In this case the left plot shows results obtained with the new lexicon and the right plot is for old lexicon models results. Speech uttered by speakers from the south of Sweden seems to be more hard to recognize, while speakers from Bergslagen give the best results. An unexpected result regards speakers from east-middle Sweden,
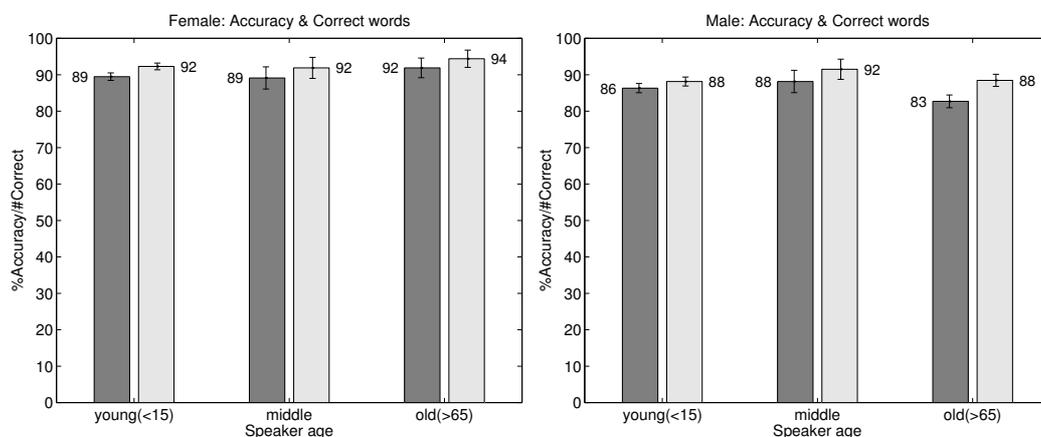
Figure 7.5. Accuracy and Correct words: sex and age distinction.
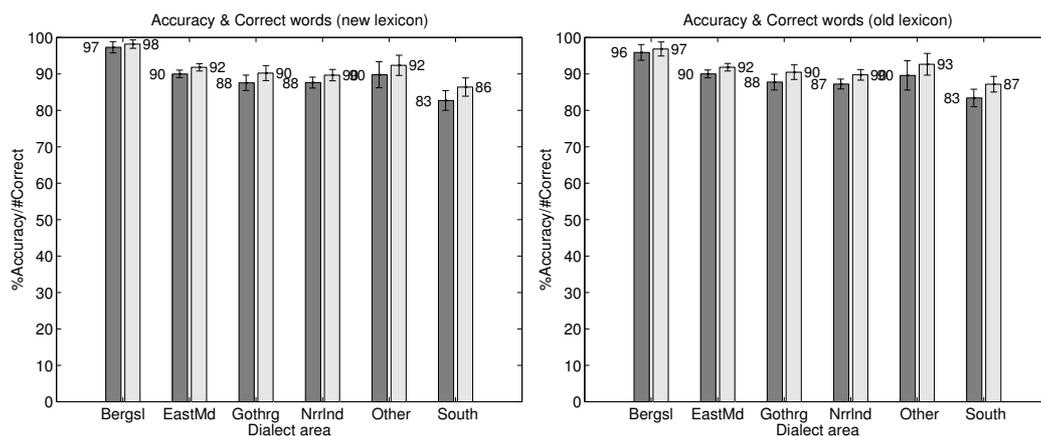
Figure 7.6. Accuracy and Correct Words: Dialect distinction.

region containing the Stockholm district, and hence the grate part of the Swedish population. In spite of the large amount of training data, for these speakers results are not as high as we would expect. The last comment on Figure 7.6 regards the lexicon. The same trend in per dialect results is obtained including or excluding retroflex allophones. In the case of southern speakers we would expect lower results for models including the retroflex allophones because speakers living in this region do not make the distinction between normal and retroflex tongue position. In spite of this results are quite similar for southern speakers, new or old lexicon.

7.4 Evaluating on Other Databases

To judge system performance it would be important to compare results with those obtained with other systems in similar conditions. In our case it is not possible to refer to a previous work on the Swedish SpeechDat database. Comparison is made referring to two experiments which contain substantial differences from the SpeechDat context. For these differences an evaluation of the model flexibility is possible. The two experiments taken into account refer to the Waxholm project and to the Norwegian SpeechDat project and will be described in the next sections.

7.4.1 The Waxholm Database

The Waxholm database presents many differences if compared to the SpeechDat database. First of all it was developed in the Waxholm project in the attempt to create a dialogue system for information about boat traffic, restaurants and accommodations in the Stockholm archipelago. The corpus of sentences included is hence affected by this task. The number of speakers (mostly men) is low if compared to the SpeechDat collection. 50% Speakers have an accent typical of the Stockholm area. Speech sampling and labeling are also different (16kHz and phone level by-hand transcriptions). Down-sampling audio files has been necessary because models developed in this work are built for telephone speech. Doing this part of the spectral information in the speech files has been lost.

Models are then tested on the same subset of ten speakers used in [16] on a generic word recognition task, even if model parameters have been tuned in this work with reference to a digit and natural number recognition task. Within-word context expansion models with eight Gaussian distributions per mixture scored 89.9% of accuracy, while in [16] 86.6% of accuracy was reached with sixteen Gaussian distributions triphones.

This prove how these models, in spite of the task adopted in this work, can be employed in a more wide range of applications.

7.4.2 The Norwegian SpeecDat database

In Norway similar experiments to those made in this work have been done for Norwegian in the SpeechDat project [8]. Results are not directly comparable for example because in our case the same network (loop of words, bigram) has been used for a wide range of different items including for example isolated digits for which a grammar definition allowing only one word (digit) for utterance would be more efficient. However, these results are always similar, even though Norwegian models have been trained on 816 speakers instead of the 750 in our experiments, and the complete database corpus has been used, while only phonetically rich sentences and words have been used in our experiments.

Chapter 8

Discussion and Conclusions

Overall results on the evaluation material have shown that models excluding ret-roflex allophones in general perform better than models including them. This con-clusion is surely affected by the task, in fact only a few words included in the recognition task (fyrtio, fjorton, arton, kontokort) contain the allophone 2T, and there is no occurrence of the other allophones. This means that splitting these mod-els in normal and retroflex versions (as in the new lexicon) results only in a lower amount of data for the normal (non retroflex) models mostly used in this task.

Furthermore models including only within-word contexts seem to perform better than models including cross word context information. This result is also affected by the task, first because one of the items included in the evaluating material consists of isolated digits (no cross-word context is available), and second because when uttering digits and natural numbers speakers tend to separate each word to make the sequence clear. In a generic speech recognition task in which a higher number of words is involved and sentences are uttered in a more continuous way, probably the advantage of using cross-word context expansion would be higher.

Per speaker results have shown how models fit quite well to different classes of speakers, with some exceptions. Finally testing models on the Waxholm database has shown the flexibility of these models in spite of the simple task they have been built for.

8.1 Further Improvements

Results for monophone models showed a considerable improvement when passing from four to eight Gaussian terms. For this reason it is likely that further im-provements are still possible adding more Gaussian terms. In the case of context dependent models this possibility seems to be more problematic, depending on the fact that the amount of data is not sufficient to train the large number of para-meters included in these models. An attempt to reduce this problem could be the use of a model set with a lower number of states respect to the optimal value (tree clustering threshold optimization) as a base to add Gaussian distributions. The

Tree clustering threshold optimization in fact is worked out on single distribution models, and there is no reason to think that the number of states in the model set is optimal also when increasing the number of Gaussian parameters. One experiment in this direction has been tested without good results.

However, when the full 5000 speakers database will be avaliable, the data scarcity problem will be reduced allowing the use of more complex models (more Gaussian distributions, lower number of states to be clustered to reduce the number of parameters). Using this database, different strategies will be possible, as for example the creation of two different model sets for female and male speakers, or the creation of dialect dependent models (for example particular models could be built for speakers from the south, which seem to have different characteristics from all the others in Sweden).

Bibliography

[1] Enrico Bocchieri and Brian Mak. Subspace distribution clustering for continuous observation density hidden markov models. In Kokkinakis et al. [10], pages 107–110.

[2] G. Chollet, F. Tore Johansen, and ... Test set definition and specification. Technical Report LE2-4001-SD1.3.4, Consortium and CEC, dec 1997.

[3] Robert Edward Donovan. *Trainable Speech Synthesis*. PhD thesis, Cambridge University Engineering Department, Trumpington Street Cambridge CB2 1PZ England, 1996.

[4] Kjell Elenius and Johan Lindberg. *FIXED1SV-FDB1000: A 1000 Speaker Swedish Database for the Fixed Telephone Network*. Department of Speech Music and Hearing KTH, SE 100 44, Stockholm, Sweden, mar 1998.

[5] Gunnar Fant. *Speech Sounds and Features*. The MIT Press Cambridge, Massachusetts and London, England, 1973.

[6] Alexandre Girardi, Harald Singer, Kiyohiro Shikano, and Satoshi Nakamura. Maximum likelihood successive state splitting algorithm for tied-mixture hmnet. In Kokkinakis et al. [10], pages 119–122.

[7] F. Jelinek. Self-organized language modeling for speech recognition. In *Readings in Speech Recognition*, pages 450–506.

[8] Finn Tore Johansen. Flexible vocabulary speech recognition for the norwegian speechdat database.

[9] Jr. John R. Deller, John G. Proakis, and John H. L. Hansen. *Discrete-Time Processing od Speech Signals*. Macmillian Publishing Company, 866 Third Avenue, New York, New York 10022, 1993.

[10] G. Kokkinakis, N. Fakotakis, and E. Dermatas, editors. *EuroSpeech 97 Proceedings: ESCA 5th European Conference on Speech Communication and Technology*, volume 1–5, 1998.

[11] Chin-Hui Lee and Lawrence R. Rabiner. A frame-syncronous network search algorithm for connected word recognition. *IEEE Transaction on Acoustics, Speech, and Signal Processing*, 37:1649–1658, November 1989.

[12] Håkan Melin. On word boundary detection in digital-based speaker verification. In *La Reconnaissance du Locuteur et ses Applications Commerciales et Criminalistiques*, pages 46–49, 1998.

[13] H. J. Nock, M. J. F. Gales, and S. J. Young. A comparative study of methods for phonetic decision-tree state clustering. In Kokkinakis et al. [10], pages 111–114.

[14] Lawrence R. Rabiner and Ronald W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall International, Inc, Englewood Cliffs, New Jersey 07632, 1978.

[15] F. Senia, R. Comeyne, Borge Lindberg, and ... Environmental and speaker specific coverage for fixed networks. Technical Report LE2-4001-SD1.2.1, Consortium and CEC, feb 1997.

[16] Kåre Sjölander. Continuous speech recognition with hidden markov models. Master's thesis, Kungliga Tekniska Högskolan Department of Speech, Music and Hearing, Drottning Kristinas väg 31 100 44 Stockholm, 1996.

[17] S. J. Young. A general use of tying in phoneme-based hmm speech recognizers. In *Proceedings of the 1992 IEEE International Conference on Acoustic, Speech and Signal Processing*, volume 1–6, pages 569–572, 1992.

[18] Steve Young, Julian Odell, Dave Ollason, Valtcho Valtchev, and Phil Woodland. *The HTK Book*. Entropic Cambridge University Laboratory, dec 1997.

[19] Victor W. Zue. From signals to symbols to meaning: on machine understanding of spoken language. In *XIIth International Congress of Phonetic Siences*, 1991.

Appendix A

Speech Signal Models

This appendix describes speech models and gives a motivation for the use of mel-cepstral analysis. In Section A.1 models for speech production are discussed, while Sections A.2 and A.3 describe cepstral analysis and the mel-frequency filter bank.

A.1 Speech Signal Models

A model for speech production consists essentially of a slowly time-varying linear system excited by either a quasi-periodic impulse train or by random noise [14]. A short segment of speech signal can hence be considered to be a stationary process and can be written as

$$
\begin{aligned}
s(n) &= p(n) * g(n) * v(n) * r(n) = p(n) * h_v(n) = \\
&= \sum_{r=-\infty}^{+\infty} h_v(n - rN_p)
\end{aligned}
$$

for voiced speech, and

$$
s(n) = u(n) * v(n) * r(n) = p(n) * h_u(n)
$$

for unvoiced speech, where $p(n)$ is a periodic impulse train of N_p samples, $h_v(n)$ is the impulse response of the linear system that combines the effects of the glottal wave shape $g(n)$, the vocal tract impulse response $v(n)$ and the radiation impulse response $r(n)$, $u(n)$ is a random noise excitation and finally $h_u(n)$ is the impulse response of the system combining the effects of the vocal tract and the radiation. In the z-transform domain

$$
\begin{aligned}
H_v(z) &= G(z)V(z)R(z) \\
H_u(z) &= V(z)R(z)
\end{aligned}
$$

are the speech system transfer functions for voiced and unvoiced segments.

A physiological study of the glottal system and the vocal tract based on pressure tube theory leads to an approximated model for these transfer functions: the vocal tract filter can be modeled by

$$V(z) \; = \; \frac{Az^{-M} \displaystyle\prod_{k=1}^{M_i} \left(1 - a_k z^{-1}\right) \prod_{k=1}^{M_o} \left(1 - b_k z\right)}{\displaystyle\prod_{k=1}^{N_i} \left(1 - c_k z^{-1}\right)}$$

For voiced speech, except nasals, an adequate model includes only poles ($a_k = 0$; $b_k = 0 \; \forall k$). As described in Chapter 2, nasal sounds are produced with a total constriction in the oral tract. The oral tract works as a resonant cavity trapping energy at natural frequencies and hence introducing zeros in the transfer function. The radiation effects result in a high frequency emphasis which can be roughly modeled by

$$R(z) \approx 1 - z^{-1}$$

Finally, for voiced speech, the glottal pulse shape is of finite duration and, thus, $G(z)$ has the form

$$G(z) = B \prod_{k=1}^{L_i} \left(1 - \alpha_k z^{-1}\right) \prod_{k=1}^{L_o} \left(1 - \beta_k z\right)$$

In general the global speech signal in the z-transform domain[1], contains slow variations (vocal tract filtering) combined with fast variations (glottal wave) [9] as depicted in figure A.1(a) (left plot).

A.2   Cepstral Analysis

An intuitive way of understanding how cepstral analysis can be useful in this context is to consider linear filtering techniques.

Problems in which linear filtering is employed usually regard a number of signals mixed together in a linear combination. The aim is usually to separate these components. For example if we consider a signal affected by addictive noise in the form

$$s = m + e$$

and we want to remove the undesired component e, we would probably need to design a system whose transfer function obeys:

$$
\begin{aligned}
y_1 &= L[m] \approx m & \text{(A.1)} \\
y_2 &= L[e] \approx 0 & \text{(A.2)}
\end{aligned}
$$

---

[1]For what we said a standard frequency domain analysis is not possible for the speech signal. In the following discussion the z-transform must be intended as a short term transform.
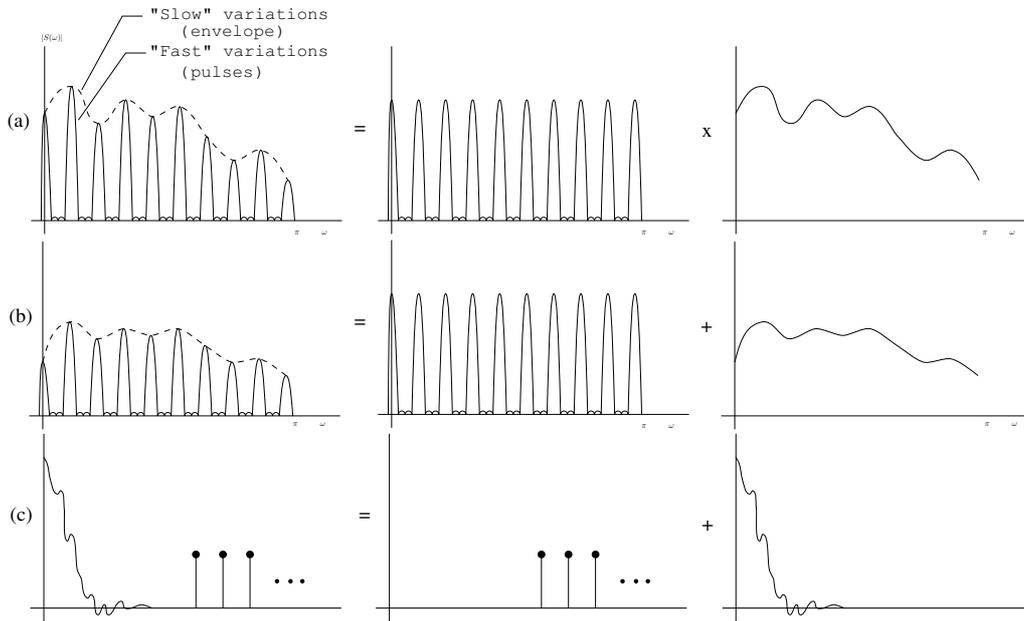
Figure A.1. Illustration of cepstral analysis of a voiced speech sound

Properties of linear systems ensure that $y = L[s] \approx m$ as wanted. The use of Fourier analysis and the introduction of a *frequency domain* help understanding and designing the filter L. Referring to the *frequency domain* we understand how the difficulty of the task depends on how well m and e are separated in frequency i.e. they occupy different frequency intervals.

The speech problem presents many differences. We still want to separate different components of the signal, but they are mixed in a convolutive way, preventing the use of standard linear filtering.

The idea is to find a new domain in which these components, or at least their representatives, are simply added. As known the spectrum of the convolution of two signals is the product of the spectra of each signal (see Figure A.1 (a)). Furthermore, if we think of the properties of logarithm, we can write

$$\log H_v(z) = \log G(z) + \log V(z)R(z) \tag{A.3}$$

as depicted in Figure A.1(b). The domain we were looking for has been called "cepstrum domain" by its first discoverers, and the transformation is intuitively defined by A.3. In this domain we can employ standard linear filtering if the aim is to remove undesired components of the signal, as in the case of Equations A.2[2]. The need to find an inverse transform to return into the time domain, as in case

---

[2]Of course the possibility of separating the two components in the new domain lies in the eventuality that they are still high and low *quefrency* even though we have subjected them to this non-linear transformation.

of Equations A.2, implies the use of a complex cepstrum (CC) which is defined as the complex logarithm of the complex spectrum. In speech applications, however, information of interest is usually contained in the cepstrum and there is no need to return into the time domain. For this reason a real cepstrum defined as the inverse fourier transform of the spectrum absolute part

$$c(n) = F^{-1}\log|Fs(n)| = \frac{1}{2\pi}\int_{-\pi}^{\pi}\log|S(\omega)|e^{j\omega n}d\omega$$

is usually employed (Figure A.1(c)). Indeed the only difference between RC and CC is that RC discards phase information about the signal while CC retains it. Although the preservation of phase bestows certain properties that are lost with the RC, the CC is often difficult to use in practise.

A.3   Mel Frequency

To understand the meaning of the *mel-frequency* filter-bank we first have to understand the term "mel". A mel is a unit of measure of perceived pitch or frequency of a tone. Its introduction is due to studies in the field of psychoacoustic. The experiment which determined the introduction of the mel scale consisted of an auditory perception test: the frequency of 1000 Hz has been chosen arbitrarily and designed as "1000 mels". Then listeners were asked to change the physical frequency until the pitch they perceived was twice the reference, 10 times, half, $\frac{1}{10}$ the reference, and so on. These pitches were labeled as 2000 mels, 10000 mels, 500 mels, 100 mels, and so on. In the end the investigators were able to say that the mapping between the real frequency scale (Hz) and the perceived frequency scale (mels) is approximately linear below 1 kHz and logarithmic above. The definition we used for the mel scale is

$$Mel(f) = 2595\log_{10}[1 + \frac{f}{700}]$$

An approach to evaluating mel-cepstrum coefficients could be oversampling the frequency axis when computing the stDFT (stFFT) and then selecting those frequency samples that represent (approximately) the mel-scale distribution, discarding all the other samples. But other considerations on human perception come in help to choose a different method: the ability of perceiving a frequency f_0 is influenced by energy in a critical band of frequencies around f_0, and the critical bandwidth varies with the frequency, being constant for $f_0 < 1$ kHz and increasing logarithmically for $f_0 > 1$ kHz. This means that we resolve better low frequencies that high ones. Therefore, rather than simply using the mel-distributed log magnitude frequency components to compute the mel-cepstrum coefficients, the log total energy in critical bands is employed. For this aim the stDFT components are weighted by filter transfer functions depicted in figure A.2 and summed to generate the log total energy for each mel frequency sample. Indeed the concatenation of DFT and sums results in a filter-bank whose transfer function is analogous to the one depicted in figure A.2.
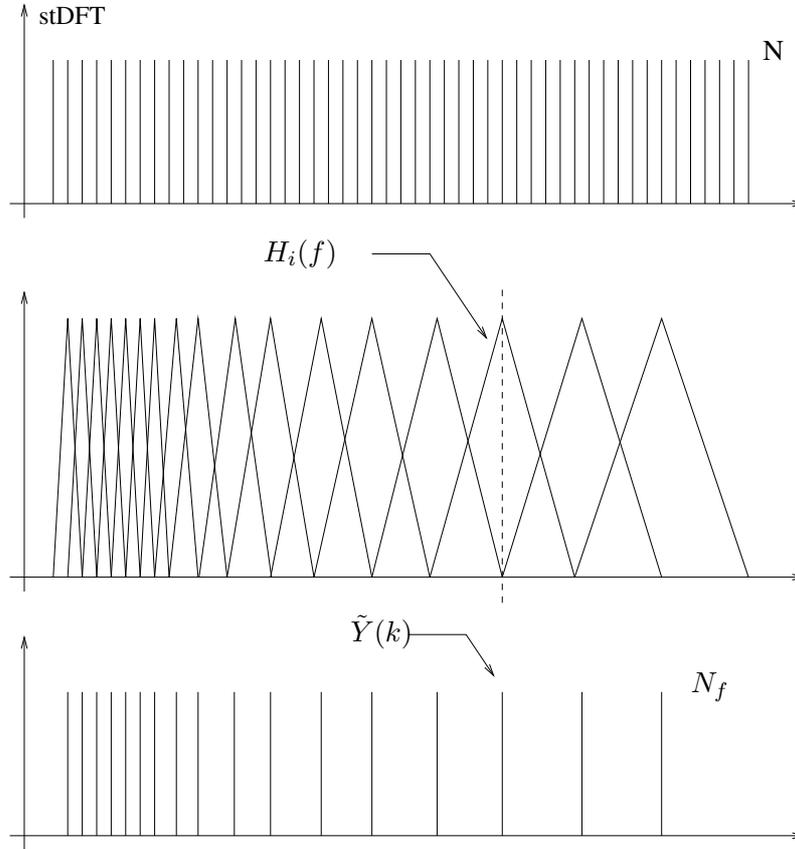
Figure A.2. Mel frequency filter bank

For a formal description of this process, the number of points in the stDFT is considered to be N and each critical band is supposed to be centered on one of the frequencies resolved by the stDFT. If we call $F_c(k)$ the center critical band frequency for the kth filter, and F_s the sample frequency in the stDFT, we can write:

$$F_c(k) = k\frac{F_s}{N}$$

Since the relation between linear frequencies and mel frequencies is not linear, only a few values of $k \in [1, N]$ are considered as shown in the upper and lower plots in Figure A.2. Let us focus attention on one of this critical filters corresponding to $k = k_i$. If we denote with $Y(i)$ the log energy output of the ith filter, and if we define

$$\tilde{Y}(k) = \begin{cases} Y(i) & k = k_i \\ 0 & k \in [1, N], \ k \neq k_i \end{cases} \tag{A.4}$$

As one the corresponding sample in the lower plot in Figure A.2, the IDFT (Inverse Discrete Fourier Transform) needed to obtain the mel-cepstrum coefficients can be

written as

$$c(n;m) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{Y}(k) \exp jk(\frac{2\pi}{N})n$$

Since $\tilde{Y}(k)$ is symmetrical about $\frac{N}{2}$ ("even") we can replace the exponential by a cosine.

$$c(n;m) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{Y}(k) cos(k\frac{2\pi}{N}n)$$

For the symmetry of $\tilde{Y}(k)$, $c(n;m)$ can be written as

$$c(n;m) = \frac{2}{N} \sum_{k=1}^{(N/2)-1} \tilde{Y}(k) cos(k(\frac{2\pi}{N})n)$$

in the assumption that N is even and that $\tilde{Y}(0) = \tilde{Y}(N/2) = 0$. The sum is extended to N terms, but referring to Equations A.4 and to Figure A.2 we see how there are only N_f nonzero terms in this sum, where N_f is the number of filters in the filter-bank.

Appendix B

HMM Theory

In this appendix problems related to HMM applications are analyzed in a more exhaustive way. Particular attention is put on computing algorithms employed in the ASR. The reason why HMM methods are still preferred to other younger theories lies in the fact that well tested computation saving algorithms such as the Forward-Backward algorithm (Section B.3), the Viterbi algorithm (Section B.4) and the Baum-Welsh re-estimation algorithm (Section B.5 exist. First of all the HMM definition is reported for clearness.

B.1   HMM Definition

An hidden Markov model (HMM) is a finite state machine which may be described at any time as being in one of a set of a finite or countable number N of possible states, $s_1, s_2, ..., s_N$. At regularly spaced discrete times, the system changes state (possibly back to the same state), and generates an output. State transitions and output generation are ruled by two stochastic processes. If we denote any time instants in which a state change is performed as $t = 0, 1, 2, ...$ and the actual state at time t as x_t, a description of such a model is given by definition of transition probability as:

$$Prob\{x_{n+1} = s_j | x_n = s_i, x_{n-1} = s_{i_{n-1}}, ..., x_1 = s_{i_1}, x_0 = s_{i_0}\}$$

For our purpose we consider only first order discrete Markov chains for which the conditional distribution of any future state x_{n+1} given the past $x_0, x_1, ..., x_{n-1}$ states and the present state x_n is independent of the past states and depends only on the present state, i.e.:

$$Prob\{x_{n+1} = s_j | x_n = s_i, x_{n-1} = s_{i_{n-1}}, ..., x_1 = s_{i_1}, x_0 = s_{i_0}\} =$$
$$= Prob\{x_{n+1} = s_j | x_n = s_i\}$$

Furthermore we suppose that whenever the process is in state s_i, there is a fixed probability that it will next be in state s_j, i.e.:

$$Prob\{x_{n+1} = s_j | x_n = s_i\} = a_{ij}(n) = a_{ij}$$

where the state transition coefficients obey standard stochastic constrains:

$$a_{ij} \geq 0$$

$$\sum_{j=1}^{N} a_{ij} = 1$$

An HMM is hence defined by:

1. a set of N reachable states $S = \{s_1, s_2, ..., s_N\}$

2. a set of M possible output symbols $V = \{v_1, v_2, ..., v_M\}$

3. a state transition probability distribution $A = \{a_{ij}\}$ where

$$a_{ij} = Prob\{x_{t+1} = s_j | x_t = s_i\}$$

4. the observation symbol probability distribution in state s_j, $B = \{b_j(k)\}$ where

$$b_j(k) = Prob\{O_t = v_k | x_t = s_j\}, \forall k \in [1, M], \forall j \in [1, N]$$

5. the initial state distribution $\pi = \{\pi_i\}$ where

$$\pi_i = Prob\{x_1 = s_i\}, \forall i \in [1, N]$$

Methods described in this appendix apply to generic HMMs as depicted in Figure B.1, though in this work only left-to-right topology has been used.
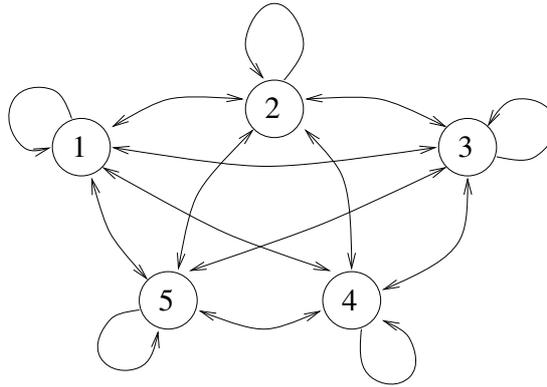
Figure B.1. A five state Markov model

B.2 Model Likelihood

As discussed in Section 5.3.1 one of the main problems in HMM applications is the computation of $Prob(O|\lambda)$. In Section 5.3.1 we said that this quantity is used in the place of $Prob(\lambda|O)$ as a measure of the likelihood that a model λ generates the (known) observation O. This assumption is valid because in the speech recognition task $Prob(\lambda|O)$ is always involved in a maximisation procedure over different models λ. For well known conditional probability properties

$$Prob(\lambda|O) = \frac{Prob(O|\lambda)Prob(\lambda)}{Prob(O)}$$

The choice of λ that maximise the right side is also the choice of λ that maximise the left size. If we assume $Prob(\lambda) = \frac{1}{K}$ (equal probability for each model) and if we consider that $Prob(O)$ is constant with λ, then maximising $Prob(O|\lambda)$ is the same that maximising $Prob(\lambda|O)$.

B.3 Forward-Backward Algorithm

The Forward-Backward (FB) algorithm has been already introduced in Section 5.3.1. In this section the forward and backward partial observation probability are formally defined:

$$\alpha_t(i) = Prob(o_1, o_2, ..., o_t, x_t = s_i | \lambda)$$

is the probability of the partial observation sequence (until time t) and state s_i at time t, given the model λ.

$$\beta_t(i) = Prob(o_{t+1}, o_{t+2}, ..., o_T | x_t = s_i; \lambda)$$

is the probability of the partial observation sequence (from time $t + 1$ to the end), given the state s_i at time t and the model λ. It is easy to see that

$$Prob(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

.

The FB procedure is based on the possibility of solving for $\alpha_t(i)$ and $\beta_t(i)$ recursively:

1. $\alpha_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N$

2. $\alpha_{t+1}(j) = \sum_{i=1}^{N} [\alpha_t(i)a_{ij}]b_j(o_{t+1})$

3. $Prob(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

Step 1 is obvious. Step 2: state s_j at time $t + 1$ can be reached from any state s_i at time t. Since $\alpha_t(i)$ is the probability of the joint event that $o_1, o_2, ..., o_t$ are observed and state s_i is reached at time t, the product $\alpha_t(i)a_{ij}$ is the probability

that $o_1, o_2, ..., o_t$ are observed and state s_j is reached at time $t + 1$ via state s_i at time t. If we sum over i we obtain the probability that $o_1, o_2, ..., o_t$ are observed and state s_j is reached at time $t + 1$???. To evaluate $\alpha_{t+1}(j)$ is hence sufficient to multiply this last quantity by $b_j(o_{t+1})$.

Step 3 gives the desired value of $Prob(O|\lambda)$ as the sum of the terminal forward variables $\alpha_T(j)$ being $\alpha_T(j) = Prob(o_1, o_2, ..., o_T, x_T = s_i|\lambda)$.

B.4   Viterbi algorithm

To describe the Viterby algorithm some quantities must be defined:

$$\delta_t(i) = \max_{q_1, q_2, ..., q_{t-1}} Prob(q_1, q_2, ..., q_{t-1}, x_t = s_i, o_1, o_2, ..., o_t|\lambda)$$

that is the highest probability along a single path at time t, considering the first t observations and ending in state s_i. By induction:

$$\delta_{t+1}(j) = [\max_i \delta_t(i)p_{ij}]b_j(o_{t+1})$$

Since in the end we want a state sequence, we need an array $\phi_t(j)$ where to store such a sequence step by step. The complete procedure is:

1. Initialization

$$\begin{aligned}
\delta_1(i) &= \pi_i b_i(o_1), 1 \leq i \leq N \\
\phi_1(i) &= 0
\end{aligned}$$

2. Recursion

$$\begin{aligned}
\delta_t(j) &= \max_{1 \leq i \leq N} [\delta_{t-1}(i)p_{ij}]b_j(o_t), 2 \leq t \leq T \\
&\quad 1 \leq j \leq N \\
\phi_t(j) &= arg \max_{1 \leq i \leq N} [\delta_{t-1}(i)p_{ij}]
\end{aligned}$$

3. Termination:

$$\begin{aligned}
P^* &= \max_{1 \leq i \leq N} [\delta_T(i)] \\
q_T^* &= arg \max_{1 \leq i \leq N} [\delta_T(i)]
\end{aligned}$$

4. Path (state sequence) backtracking:

$$q_t^* = \phi_{t+1}(q_{t+1}^*), t = T - 1, T - 2, ..., 1$$

B.5   Baum-Welsh Reestimation Algorithm

The aim is to adjust the model parameters (A, B, π) to maximise the probability of the observation sequence, given the model. This is an NP-complete problem, and hence has no analytical solution. In fact, given any finite observation sequence as training data, we can only try to choose $\lambda = (A, B, \pi)$ such that $Prob(O|\lambda)$ is locally maximised. There are many methods for doing it, among which gradient techniques, expectation-modification technique, genetic algorithms. In this section we describe an iterative procedure called Baum-Welch method. First of all we define $\xi_t(i, j)$, the probability of being in state s_i at time t and in state s_j in time $t + 1$, given the observation sequence and the model:

$$\xi_t(i, j) = Prob(x_t = s_i, x_{t+1} = s_j | O, \lambda)$$

We can write this quantity in function of the forward and backward variables; according to their definitions:

$$
\begin{aligned}
\xi_t(i, j) &= \frac{\alpha_t(i) p_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{Prob(O|\lambda)} \\
&= \frac{\alpha_t(i) p_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\displaystyle\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) p_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}
\end{aligned}
$$

Defining $\gamma_t(i)$ as the probability of being in state s_i at time t, given the observation sequence and the model we can obtain this probability by summing $\xi_t(i, j)$ over j:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j)$$

Furthermore if we sum $\gamma_t(i)$ over time $(1 \leq t \leq T)$ we get the expected (over time) number of times state s_j is visited. Similarly if we sum $\xi_t(i, j)$ over $1 \leq t \leq T - 1$ we get the expected number of times transition from state s_i to state s_j is performed, and summing $\gamma_t(i)$ over $1 \leq t \leq T - 1$ the expected number of times transition from state s_i to any other state is performed. We are now ready to give the estimation procedure: suppose that we have a randomly chosen, or badly trained, model $\lambda = (P, B, \pi)$, we evaluate $\gamma_t(i)$ and $\xi_t(i, j)$ using this model and the training data (observation sequences). Than we estimate new parameters $\hat{\lambda} = (\hat{P}, \hat{B}, \hat{\pi})$ as following:

$$
\begin{aligned}
\hat{\pi} &= \text{number of times in state } s_i \text{ at time } t = 1 \\
\hat{p}_{ij} &= \frac{\text{number of transitions from state } s_i \text{ to state } s_j}{\text{number of transitions from state } s_i} \\
\hat{b}_j(k) &= \frac{\text{number of times in state } s_j \text{ and observing symbol } v_k}{\text{number of times in state } s_j}
\end{aligned}
$$

according to the definitions of $\gamma_t(i)$ and $\xi_t(i,j)$ we have:

$$\hat{\pi} = \gamma_1(i)$$

$$\hat{p}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(j)}$$

$$\hat{b}_j(k) = \frac{\sum_{t=1,o_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Sums in the ?? equation are extended from $t = 1$ to $t = T - 1$ for obvious reasons. Baum proved that two cases are possible:

1. the initial model λ defines a critical point in likelihood function (null gradient), and in this case $\hat{\lambda} = \lambda$

2. $Prob(O|\hat{\lambda}) > Prob(O|\lambda)$ and hence we have found a new model for which the observation sequence is more likely.

If we repeat this procedure using iteratively $\hat{\lambda}$ in place of λ, we should be able to reach the nearer critical point in the likelihood function.

Appendix C

Lists Of Words

In this appendix lists of words used in recognition (developing and testing) are included. In table C.1 digits (0-9) are listed, while table C.2 shows natural numbers. Some of them have one or more variants in pronunciation. The other words are listed in table C.3, including noise symbols and sentence boundary symbols.

English	Swedish	Variants	
zero	noll	nolla	
one	ett	en	etta
two	två	tvåa	
three	tre	trea	
four	fyra		
five	fem	femma	
six	sex	sexa	
seven	sju	sjua	
eight	åtta		
nine	nio	nia	nie

Table C.1. Digits

10..19	Variants	20..90	Variants	100, 1000, 10^6	Variants
tio	tie	tjugo	tju, tjugi, tjugu	hundra	
elva		tretti	trettio	tusen	
tolv		fyrti	fyrtio	miljon	miljoner
tretton		femti	femtio		
fjorton		sexti	sextio		
femton		sjutti	sjuttio		
sexton		åtti	åttio		
sjutton		nitti	nittio		
arton					
nitton					

Table C.2. Natural Numbers

bindestreck	det	eller	en	har	här
inget	inte	jag	jämt	komma	kontokort
krona	kronor	men	nej	nummer	och
parantes	som	spänn	streck	står	va
vad	vet	är	öre	[fil]	[int]
!ENTER	!EXIT				

Table C.3. Other words

Appendix D

Phonetic Classes

The decision tree, on which the clustering procedure is based, is a collection of questions regarding the classification of the left and right context of each phone. In this appendix the phonetic classification used in this work is reported. Each class in the list below corresponds to a possible question in the clustering procedure.

CENTRAL	U, E0
APPROXIMANT	V, J, R
LATERAL	L
LABIODENTAL	V, F
BACK-MIDHIGH	Å:, Å
PALATAL	J, TJ
LOW	A:, A
BACK-HIGH	O:, O
BILABIAL_PLOSIVE	B, P
VELAR_PLOSIVE	G, K
NASAL	M, N, NG
BILABIAL	B, P, M
BACK-ROUNDED	O:, O, Å:, Å
RETROFLEX	_2S
VELAR	G, K, NG
FRICATIVE	F, S, _2S, TJ, SJ, H
FRONT-HIGH	I:, I, Y:, Y, U:
VOICED_PLOSIVE	B, D, G
UNVOICED_PLOSIVE	P, T, K
SPDAT_PLOSIVE	B, D, G, K
BACK	A:, O:, O, Å:, Å
FRONT_ROUNDED	U:, Y:, Y, Ö3, Ö4, Ö:, Ö
CORONAL	D, L, N, R, S, _2S, T
ALVEOLAR	D, T, S, N, L, R
HIGH	I:, I, O:, O, U:, Y:, Y
CONTINUANT_CONSONANT	V, F, S, L, R, J, TJ, SJ, H

FRONT_UNROUNDED	A, E:, E, I:, I, Ä3, Ä4, Ä:, Ä
ROUNDED	O:, O, U:, U, Y:, Y, Å:, Å, Ö3, Ö4, Ö:, Ö
UNVOICED_CONSONANT	P, T, S, _2S, TJ, K, SJ, H
FRONT_MIDHIGH	E:, E, Ä3, Ä4, Ö3, Ö4, Ä:, Ä, Ö:, Ö, E, E0
PLOSIVE	B, P, D, T, G, K
ANTERIOR	B, P, V, F, M, D, T, S, N, L
MIDHIGH	E:, E, U, Å:, Å, Ä3, Ä4, Ö3, Ö4, Ä:, Ä, Ö:, Ö
VOICED_CONSONANT	B, V, M, D, N, L, R, J, G, NG, H
OBSTRUENT	B, P, F, D, T, S, _2S, J, TJ, G, K, SJ, H, V
FRONT	A, E:, E, I:, I, U:, Y:, Y, Ä3, Ä4, Ö3, Ö4, Ä:, Ä, Ö:, Ö
VOWEL	A:, A, E:, E, I:, I, O:, O, U:, U, Y:, Y, Å:, Å, Ä3, Ä4, Ö3, Ö4, Ä:, Ä, Ö:, Ö, E0
VOICED	A:, A, E:, E, I:, I, O:, O, U:, U, Y:, Y, Å:, Å, Ä3, Ä4, Ö3, Ö4, Ä:, Ä, Ö:, Ö, E0, B, V, M, D, N, L, R, J, G, NG, H
E_VOCALS	E:, E
I_VOCALS	I:, I
Y_VOCALS	Y:, Y
Ö_VOCALS	Ö3, Ö4
Ä_VOCALS	Ä3, Ä4
HIGH_ROUNDED	Y:, Y, U:
MIDHIGH_ROUNDED	Ö:, Ö, Å:, Å, Ö3, Ö4
MIDHIGH_PRE_R	Ö3, Ö4, Ä3, Ä4
MIDHIGH_UNROUNDED	E:, E, E0, Ä3, Ä4
TENSE	A:, E:, I:, O:, U:, Y:, Å:, Ä3, Ö3, Ä:, Ö:

Appendix E

Confusion Matrices

In this appendix (next pages) confusion matrices are included for each experiment: monophones/triphones, within/cross-word context expansion, old/new lexicon. Models in each model set include eight Gaussian mixture terms as state to output probability distribution.

```
---------------------------- Confusion Matrix ----------------------------
       n  e  t  t  t  f  f  s  s  å  n  t  e  t  t  f  f  s  s  å  n  t  t  f  f  s  s  a  n  h  t  m  d  e  h  h  i  j  k  k  m  n  o  s  s  s  s  v  v  ä  ö
       o  t  v  r  y  e  e  j  t  i  i  l  o  r  y  e  e  j  t  i  j  r  j  e  e  j  r  i  u  u  i  e  l  a  ä  n  a  o  r  e  e  c  o  p  t  t  a  e  r  r
       l  t  å  e  r  m  x  u  t  o  o  v  l  e  r  m  x  u  t  t  u  e  o  m  x  u  t  t  n  s  l  t  l  r  r  t  g  m  o  n  j  h  m  ä  r  å  d  t        e
       l           a           a           a  v  t  t  t  t  t  t  i  t  g  t  r  t  t  t  t  o  t  d  e  j        e              a  o           n  c
                            t  i  i  i  t  o  i  o  t  t  o  o  t  n  o  r  n  o        r                          n
                                                                                                                                              Del [ %c / %e]
noll 874  6 50  2  0  0  0  1  4  7  0  0  1  1  0  2  2  3  3  4  1  2  1  8  7  0  2  0  0  0  0  1  0  3  1  5  5  2  4  1  0  5  0  0  0  0  1  0  2  0   83 [86.4/1.2]
 ett   0 650 36 79  0  1 15  1  6  3  3  0  0 10  0  1  5  1  0  5  0  0  1  0  1  1  0  3  1  0  1  0  0  2  0  0  0  0  1  0  0  1  0  0  1  2  0  100 [78.2/1.6]
 två   1  3 770  4  0  3  0  0  5  2  1  0  1  0  0  0  2  1  2  0  1  3  2  2  1  0  1  0  0  1  0  0  0  1  0  0  0  4  0  0  3  1  0  0  1  0  0  0  0  0   73 [94.4/0.4]
 tre   1  4  2 667 15  3  2  8  1  8 25  0  2  5  0  4 10  0  5  3  3  1  1  0  0  1  0  0  1  2  0  5  0  0  0  0  0  0  1  0  1  1  0  0  0  0  0  2  0   89 [85.1/1.1]
fyra   0  1  3  5 742  2  1  2  1  7  6  1  0  0  1  2  0  0  0  0  4  0  0  0  0  0  0  2  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0   57 [95.0/0.4]
 fem   0 56  4  4  7 613  2  0  7  1  1  2  0  0  0  7  0  1  1  1  1  0  0  5  2  2  0  0  0 17  1  0  0  1  1  0  0  1  1  0  2  0  2  0  0  0  1  0   89 [82.5/1.2]
 sex   2 11  0  1  0  2 661  0  1  1  1  0  0  0  0  0 22  0  0  1  2  1  0  0  7  0  1  0  2  0  0  0  0  0  0  0  4  0  0  1  0  0  0  0  0  0  0   81 [91.7/0.5]
 sju   4  3  3  4  6  3  0 625  2  0  0  0  1  0  0  1  0  3  0  1  4  0  0  1  1  1  0  1  2  3  0  0  0  0  2  0  0  1  0  0  0  1  0  0  0  1  0   73 [92.7/0.4]
åtta   0  2  7  5  1  0  0  0 639  0  0  0  1  0  0  0  3  0  2  1  1  1  1  0  1  0  0  1  0  0  1  0  0  0  0  1  0  0  0   67 [95.9/0.2]
 nio   5 12  3 21 17  0  0  1  0 527 12  0  0  0  0  3  2  0  1  5  4  0  4  0  0  0  1  0  0  3  2  1  0  0  2  4  0  6  0 10  1  0  0  0  0  0   82 [81.5/1.1]
 tio   1  1  0  4  2  0  0  0  0  5 43  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0    5 [75.4/0.1]
elva   0  3  0  0  0  0  0  0  1  0  0 59  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0    3 [90.8/0.1]
tolv   4  2  9  0  0  0  0  0  0  0  0  0 20  0  0  0  0  0  0  0  0  0  3  0  1  2  2  2  0  0  0  0  1  0  0  1  0    4 [42.6/0.2]
tret   0  5  1  7  0  0  2  2  1  2  0  0  0 116  0  0  2  1  0  1  0  0  0  0  0  0  0  0  0  1  0    14 [83.5/0.2]
fyrt   0  1  0  4  0  1  1  1  0  0  0  0  0  0 88  0  1  0  0  0  0  0  0  1  0  0    8 [89.8/0.1]
femt   1  5  0  1  0  8  4  0  3  2  0  0  1  9  1 85  0  1  1  1  0  0  0  0  2  0  1    17 [68.0/0.4]
sext   0  1  1  1  0 18  1  0  0  0  0  0  1  1  1 101  0  0  0  0  0  0  2  0  0  0  1    10 [?/0.3]
sjut   0  2  1  2  0  0  8  0  0  0  0  0  3  3  0  1 124  2  0  4  0  0  0  0  1    13 [?/0.2]
åtti   3  1 10 15  0  0  0  0  9  2  2  0  0  0  0  3 113  0  1  0  0  0  0  1  0  1  0  4  0  0  1  0    20 [?/0.5]
nitt   0  5  0  2  0  0  0  0  3  1  0  3  0  1  3  0  1 160  0  0  1  1  0    12 [?/0.4]
tjug   2  2  2  1  0  0  3  0  0  1  0  0  0  0  1  3  0  0 157  0  0    12 [?/0.1]
tret   0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0 33  0  0  0  1  0    1 [?/0.0]
fjor   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0    0
femt   2  0  0  0  0  0  0  0  1  0  0  1  0  1  0  0  0  0  0  0 37  0    2 [?/0.0]
sext   0  0  0  0  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0 40  0    2 [95.2/0.0]
sjut   0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  0    0 [?/0.0]
arto   0  0  0  1  0  0  0  0  2  0  0  0  1  0  0  0  0  0  0  0  0  0  0 16  0  0  0  0  0  0  1  0    1 [?/0.0]
nitt   0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  1  0 29  0  0  0  0  1  0    1 [?/0.0]
hund  13  5  3  1  6  6  1  1 14  6  2  2  3  1  0  2  0  1  1  0  3  2  1  3  0  3  1 265  6  0  0  0  0  0  1  2  0  1  0  5  0  0  0  0  1  2  0   75 [?/0.9]
tuse   0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 144  0    5 [?/0.0]
milj   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 23  0  0  0  0  0  1  0    1 [?/0.0]
komm   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    0
kron   0  0  1  0  1  0  1  1  2  0  0  0  0  0  0  0  0  1  0  0  1  1  1  0  0  1  0  0  0  0  0  0  0  0 167  0    8 [?/0.1]
 och   0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 14  0    5 [93.3/0.0]
stre   0  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  3  0    2 [?/0.0]
 öre   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  8    2
 Ins  13 88 39 63  6  8  6  4 19 15  3  0  0  3  0  0  3  1  2  3  4  1  1  0  1  0  0  0  3  6  0  7  1  1  0  2  6  2  1  1  0  6  0  0  0  0  1  0  2  0
==========================================================================
```

Figure E.1. Confusion matrix: monophones, old lexicon, eight Gaussian mixture terms

```
---------------------------- Confusion Matrix ----------------------------
      n  e  t  t  t  f  f  s  s  å  n  t  e  t  t  f  f  s  s  å  n  t  t  f  f  s  s  a  n  h  t  m  d  h  h  i  j  k  k  k  m  n  o  s  s  s  s  v  ä  ö
      o  t  v  r  y  e  e  j  t  i  l  o  r  y  e  e  j  t  i  j  r  j  e  e  j  r  i  u  u  i  e  a  ä  n  a  o  r  r  e  e  c  o  p  t  t  a  r  r
      l  t  å  e  r  m  x  u  t  o  o  v  l  e  r  m  x  u  t  t  u  e  o  m  x  u  t  t  n  s  l  t  r  r  t  g  m  o  o  n  j  h  m  ä  r  å  d     e
      l              a           a           a  v  t  t  t  t  t  t  i  t  g  t  r  t  t  t  o  t  d  e  j              e        m  n  n                 n  e  r
                                                 t  i  i  i  t  o  i  o  i  o  t  t  o  o  t  n  o  r  n  o                       a  a  o                 n  c            Del [ %c / %e]
noll 891  5 52  2  3  0  0  1  6  8  0  0  1  2  0  1  2  0  1  2  2  1  1  7 10  2  2  0  0  0  2  1  5  0  3  5  2  0  3  1  0  4  1  0  0  0  0  3  0    62 [86.3/1.3]
 ett  1 660 45 79  1  2 16  2 11  1  0  0  0  8  0  1  3  2  0  2  0  2  1  0  0  0  2  1  0  2  0  0  4  0  0  0  0  0  1  0  0  0  0  0  0  0  2  0    82 [77.7/1.7]
 två  0  2 800  0  2  5  0  0  4  0  1  0  2  1  0  0  2  0  1  0  1  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  7  0  0  0  0  0  0  0  0  0  0    59 [96.4/0.3]
 tre  1  6  1 692 21  2  2 10  0  6 17  1  0  7  0  8 12  1  0  6  0  1  0  0  1  0  0  0  2  2  1  2  0  1  0  0  0  0  0  2  1  0  1  0  0  0  1  0    65 [85.6/1.1]
fyra  1  2  2  4 756  1  1  5  3  3  2  0  1  0  1  2  1  0  0  1  0  0  1  0  0  0  5  1  0  3  0  1  1  0  0  0  0  0  1  0  0  0  0  0  0  0  1  0    38 [94.5/0.4]
 fem  0 11  3  7  7 645  6  0  4  0  2  1  1  0  0  6  1  0  1  0  0  0  6  2  1  0  0  2 38  0  0  2  2  0  1  0  0  2  2  0  1  0  4  0  1  0  2  0    71 [84.8/1.1]
 sex  1  9  0  2  0  3 703  0  0  1  1  0  0  0  0  0  9  0  0  0  2  0  0  1  4  0  0  0  2  1  0  0  0  0  0  0  0  0  1  0  1  0  1  1  0  0  1  0    59 [94.6/0.4]
 sju  4  1  2  5  8  3  0 644  0  1  0  0  0  1  0  0  0  1  0  1  4  0  0  1  1  3  0  1  2  5  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0    57 [93.3/0.4]
åtta  1  1 10  2  2  0  0  0 655  0  0  0  0  0  0  0  0  0  6  0  0  2  0  1  0  3  0  0  1  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0    46 [95.3/0.3]
 nio  5 12  5 37 48  2  0  7  4 463 10  2  0  0  0  2  1  0  0  8  2  0  3  1  1  1  2  0  2  5 11  1  0  1  1  4  0  0  3  1 19  0  0  0  0  0  1  0    64 [69.6/1.8]
 tio  0  3  0  6  3  0  0  2  0  6 33  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     7 [60.0/0.2]
elva  0  0  0  0  1  0  0  0  0  0  0 63  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     2 [95.5/0.0]
tolv  4  0  8  0  0  0  0  0  0  1  0  0 20  0  0  0  0  0  2  0  0  3  0  2  1  1  0  0  2  0  0  0  1  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0     4 [42.6/0.2]
tret  0 14  2 13  0  0  0  1  1  0  0  0  0 104  0  0  1  2  0  1  0  0  0  0  0  0  0  0  2  0  0  0  2  1  5  0  3  0  0  0  0  0  0  0  0  0  0  0    12 [73.8/0.3]
fyrt  1  1  1  3  0  1  2  0  0  0  0  0  0  1 86  1  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     6 [86.0/0.1]
femt  1  3  1  5  0 10  3  0  2  1  0  1  1  0 97  0  3  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0    12 [74.6/0.3]
sext  0  2  1  1  0 26  0  0  0  0  0  0  0  1  0 96  0  0  0  0  0  0  0  1  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     9 [73.8/0.3]
sjut  0  4  2  2  0  0  0 12  0  0  0  0  0  1  2  0  1 123  0  0  3  0  0  0  0  3  0  0  1  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0     9 [79.4/0.3]
åtti  3  3 14 26  0  0  1  0 19  3  7  0  0  1  2  0  0  2 67  3  1  0  0  0  0  0  2  0  1  1  0  1  0  1  0  0  0  0  7  0  0  0  0  0  0  0  0  0    22 [40.6/0.9]
nitt  0  7  1  8  0  0  0  6  0  0  0  1  0  3  1  1  0 149  0  0  0  0  0  0  0  2  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    13 [82.8/0.3]
tjug  2  2  1  2  1  0  4 18  0  0  1  0  0  0  0  1  1  0  0  0 123  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0    25 [77.4/0.3]
tret  0  0  1  0  0  0  0  0  0  0  0  0  0  0  2  0  0  1  0  0  0  0 31  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     1 [86.1/0.0]
fjor  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     0
femt  1  0  0  0  0  0  0  0  1  0  0  1  0  1  0  1  0  0  0  0  0  0  0  0 37  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     2 [88.1/0.0]
sext  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0 39  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     3 [95.1/0.0]
sjut  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     1 [96.7/0.0]
arto  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 19  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0     1 [90.5/0.0]
nitt  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  1  0  32  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     0 [91.4/0.0]
hund 16  3  5  4  7  8  1  1 13  2  2  2  1  0  0  0  1  2  0  0  0  1  1  1  2  3  1  1 250  3  0  1  0  0  0  1  4  0  0  0 12  0  0  0  0  3  0    87 [71.0/0.9]
tuse  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0 147  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     2 [99.3/0.0]
milj  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 23  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0     1 [95.8/0.0]
komm  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  0     0
kron  0  1  1  0  1  0  0  0  2  1  0  0  0  0  0  0  0  0  0  1  0  1  2  0  1  0  0  1  0  3  0  0  0  0  0  0  1  1 162  0  0  1  0  0  0  0  0  0     6 [90.0/0.2]
 och  0  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0 12  0  0  0  0  0  0  0  0  0  0  0     5 [80.0/0.0]
stre  0  0  0  1  0  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0     1 [57.1/0.0]
 öre  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  6     2 [75.0/0.0]
 Ins 14 99 45 78 11 15  9 10  9 22  3  0  1  1  0  1  1  1  1  1  1  1  0  0  0  3  1  4 11  0 13  3  0  3  7  1  0  1  3  3  8  1  0  0  0  1  4  0
==========================================================================
```

Figure E.2. Confusion matrix: monophones, new lexicon, eight Gaussian mixture terms

```
---------------------------- Confusion Matrix ----------------------------
     n e t t f f s s å n t e t t f f s s å n t t f f s s a n h t m d h i j k k m n o s s s v ä ö
     o t v r y e e j t i i l o r y e e j t i j r j e e j r i u u i e a n a o r e e c o p t a r r
     l t å e r m x u t o o v l e r m x u t t u e o m x u t t n s l t r t g m o n j h m ä r d   e
     l   a                 a   a v t t t t t t i t g t r t t t t o t d e j     e     m n       n e
                               t i i i t o i o t t o o t n o r n o           a o           n c   Del [ %c / %e]
noll 1043 2  5  0  0  0  0  1  0  2  0  0  0  0  0  0  0  0  0  0  1  0  0  1  1  0  0  3  0  0  0  0  1  1  0  0  2  0  0  2  0  0  0  0  1  0   28 [97.8/0.2]
ett   0 850  1  9  1  4  1  0  4  0  0  0  0  1  0  0  2  0  1  0  1  0  0  0  0  1  2  1  0  0  1  0  0  0  2  0  0  0  0  0  0  0    49 [96.4/0.3]
två   5  3 820  1  1  6  1  0  3  0  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1  0  0  2  0  0  0  1  0  0  1  0  0  4  0  0  2  0  0   36 [96.1/0.3]
tre   2  6  1 771 10  5  1  6  0  2  5  0  0  0  1  0  4  0  1  2  1  0  0  0  0  0  0  0  2  0  5  0  0  1  0  2  0  3  0  0  0  0  0  0   42 [92.8/0.5]
fyra  0  2  0  1 800  1  0  0  0  0  1  3  0  0  1  0  0  0  2  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0   25 [98.4/0.1]
fem   0 23  0  2  1 748  0  0  2  0  0  0  0  0  0  4  0  0  1  0  0  0  5  1  0  0  1  4  0  0  0  0  0  0  0  1  0  0  0  0  1  0   38 [94.2/0.4]
sex   0  6  0  2  0  0 741  0  0  1  0  0  0  0  1  0  0  0  0  0  1  1  0  4  0  0  0  0  0  1  0  0  0  0  0  1  0  0  1  0  0  0   42 [97.5/0.2]
sju   5  4  1  4  3  2  1 677  0  0  0  0  2  0  0  0  0  1  1  0  7  0  0  1  0  1  1  0  0  1  0  0  0  1  0  0  0  0  1  0  0  0   33 [94.8/0.3]
åtta  0  1  0  0  1  0  0  0 689  0  0  0  0  1  0  0  0  1  3  0  0  0  0  0  5  0  0  0  0  0  0  0  2  0  0  0  0  0  0  1  0   29 [97.9/0.1]
nio   7 15  0  8 20  2  0  2  0 596  5  0  0  0  0  0  0  0  1  8  1  0  0  0  0  2  2  2  1  6  5  0  1  0  0  0  3  3  0  0  0  0   39 [86.4/0.9]
tio   0  2  0  2  4  0  0  0  0  3 50  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    1 [82.0/0.1]
elva  0  1  0  0  1  0  0  0  0  0  0 63  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0  0    1 [94.0/0.0]
tolv  1  0  3  0  0  0  0  0  0  0  0  0 45  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0    0 [88.2/0.1]
tret  0  2  0  0  1  0  0  0  0  0  1  0  0 141  1  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0    3 [94.0/0.1]
fyrt  0  1  0  1  0  0  0  0  0  0  0  0  0  2 97  2  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    2 [93.3/0.1]
femt  0  2  0  1  0  4  0  0  0  1  0  0  0  1 128  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    4 [92.8/0.1]
sext  0  1  0  0  1  0  8  1  0  0  0  0  0  0  0 126  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    2 [92.0/0.1]
sjut  0  1  0  0  0  0  0  1  0  0  0  0  0  2  4  0  0 149  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0    4 [93.1/0.1]
åtti  2  1  0  2  0  0  0  0  6  0  1  0  0  0  1  0  0  2 153  0  0  0  0  0  1  0  0  0  0  0  2  0  0  3  0  0  0  0  0  0    13 [87.9/0.2]
nitt  0  1  0  1  0  0  1  0  0  0  1  0  0  0  0  2  0 177  0  0  0  0  0  2  0  0  0  0  0  1  0  0  1  0  0  0  1  0  0    5 [94.1/0.1]
tjug  4  0  0  1  0  0  1  2  0  0  0  0  0  0  0  0  1  0  0 158  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0    16 [94.0/0.1]
tret  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 35  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0    0 [94.6/0.0]
fjor  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0    0
femt  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 43  0  0  0  0  0  0  0  0  0  0  0  0    1
sext  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 42  0  0  0  0  0  0  0  0  0  0  0    2
sjut  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 31  0  0  0  0  0  0  0  0  0  0    0
arto  0  0  0  0  0  0  0  1  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 19  0  0  0  0  0  0  0  0  0    0 [86.4/0.0]
nitt  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 33  0  0  0  0  0  0  0  0    0 [94.3/0.0]
hund  8  9  1  1  4  2  0  0  0  0  0  1  2  0  0  0  0  0  0  1  0  0  7  0  3  0 334  0  0  0  0  0  2  1  0  0  5  1  0  0  0  2  0   55 [87.0/0.5]
tuse  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 146  0  0  0  0  0  0  0  0    2 [98.6/0.0]
milj  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 23  0  0  0  0  0  1  0  0  0    0 [92.0/0.0]
komm  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  6  0  0  0  0  0  0    0
kron  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0 180  0  0  0  0  1  0  0  0  0    3 [98.4/0.0]
och   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 13  0  0  0  0    7
stre  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0  0    0 [87.5/0.0]
öre   0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  8    1 [88.9/0.0]
Ins   6 120  5 18  5 11  3 14  9  6  6  1  1  1  1  1  0  0  3  0  0  1  0  0  0  0  1  0  2  1  0 10  1  3  4  0  3  0  2 16  1  0  3  0  3  0
============================================================================
```

Figure E.4. Confusion matrix: triphones, within-word context expansion, new lexicon, eight Gaussian mixture terms

```
----------------------------- Confusion Matrix -----------------------------
       n  e  t  t  t  f  f  s  s  å  n  t  e  t  t  f  f  s  s  å  n  t  t  f  f  s  s  a  n  h  t  m  d  h  h  i  j  k  k  k  m  n  n  o  s  s  s  v  ä  ö
       o  t  v  r  y  e  e  j  t  i  i  l  o  r  y  e  e  j  t  i  j  r  j  e  e  j  r  i  u  u  i  e  a  ä  n  a  o  r  r  e  e  u  c  o  t  t  a  r  r
       l  t  å  e  r  m  x  u  t  o  o  v  l  e  r  m  x  u  t  t  u  e  o  m  x  u  t  t  n  s  l  t  r  r  t  g  m  o  o  n  j  m  h  m  r  å        e
       l              a              a              v  t  t  t  t  t  t  i  t  g  t  r  t  t  t  o  t  d  e  j              e        m  n  n        m        e  r
                                                    t  i  i  i  t  o  i  o  t  t  o  o  t  n  o  r  n  o                    a  a  o        e        c      Del [ %c / %e]
noll 1045  1  4  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1  1  1  0  0  3  0  2  2  0  1  0  0  0  0  0  1  0   28 [98.0/0.2]
ett    0 829  1 12  2  4  1  4  8  0  0  0  2  0  1  1  0  1  0  1  1  0  0  0  2  0  0  1  0  0  2  0  0  0  0  0  0  0  0  1  0   57 [94.9/0.4]
två    6  3 815  0  1  8  1  1  4  0  0  0  1  0  0  0  0  0  0  1  0  1  0  0  0  0  1  0  0  0  0  1  2  0  0  1  0  0  0  0  1  0  1  0   40 [96.0/0.3]
tre    2  5  0 788  5  4  2  7  0  4  1  0  0  0  0  1  0  0  1  0  0  0  0  1  0  0  0  0  1  0  2  0  0  0  1  0  0  1  0  2  0  0  0  0  2  0   43 [94.9/0.4]
fyra   1  0  0  3 799  0  0  2  3  1  1  1  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0   24 [98.2/0.1]
fem    1  9  0  0  2 762  0  2  3  0  0  0  0  0  3  0  0  0  0  0  0  0  1  0  0  0  1  3  0  1  1  1  0  0  0  0  1  0  0  0  0  0   41 [96.3/0.3]
sex    0  8  0  2  0  0 749  0  0  0  0  0  0  0  0  0  1  0  0  0  1  2  0  0  2  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  1  0   34 [97.5/0.2]
sju    3  2  1  5  2  2  1 684  0  1  0  1  3  0  0  0  1  0  5  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  35 [96.1/0.3]
åtta   0  0  1  0  1  0  0  0 692  0  0  0  1  0  0  0  4  0  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  1  0   30 [98.4/0.1]
nio    7  6  1 11 29  1  0  5  0 602  0  1  0  0  0  0  0  0  6  0  0  0  0  0  0  3  2  5  4  0  0  2  2  0  0  3  0  0  0  0  0  0   39 [87.2/0.8]
tio    0  2  0  5  4  1  0  0  0  3 44  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    2 [73.3/0.1]
elva   0  0  0  0  1  0  0  0  0  0  0 63  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0    1 [94.0/0.0]
tolv   0  1  2  0  0  0  0  0  0  0  0  0 45  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    1 [90.0/0.0]
tret   0  7  0  2  1  0  0  0  0  0  0  0  0 137  1  0  1  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0    2 [90.7/0.1]
fyrt   0  1  0  2  0  0  1  0  0  0  0  0  0  0 92  2  1  4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    3 [89.3/0.1]
femt   1  0  0  2  0  2  0  0  0  1  0  0  0  0  1 128  0  0  0  2  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0    3 [92.1/0.1]
sext   0  2  0  3  1  0 10  1  0  0  0  0  0  0  0  0 118  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0    2 [86.1/0.2]
sjut   1  0  0  0  0  0  1  4  0  0  1  0  0  2  2  0  0 143  1  1  2  0  0  0  0  0  0  0  0  0  0  0    6 [90.5/0.1]
åtti   1  1  0  7  0  0  0  0  5  1  1  0  0  0  1  0  0  8 142  2  0  0  0  0  0  1  0  0  0  2  0  0  0  4  0  0  0  1  0  0  0  1  0    9 [79.8/0.3]
nitt   0  2  0  2  0  0  1  1  0  1  0  0  0  0  0  0  0  2  0 176  0  0  0  0  2  0  0  0  0  1  0  0  0    5 [93.6/0.1]
tjug   3  1  0  1  0  0  2  2  0  0  0  0  0  1  0  0  2  0  0  0 152  0  0  0  0  0  0  0  0  1  0  0  0   19 [92.1/0.1]
tret   0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 36  0  0  0  0  0  0  0    0 [97.3/0.0]
fjor   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0    0
femt   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 42  0  1  0    1 [97.7/0.0]
sext   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 42  0  0    2
sjut   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  1  0    0 [96.8/0.0]
arto   0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 19  0    0 [86.4/0.0]
nitt   1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 34  0    0 [97.1/0.0]
hund   9  1  1  0  2  3  0  1  1  0  0  0  2  0  0  0  0  0  0  1  0  3  2  0  1 346  0  0  1  0  0  0  4  0  1  0  0  0  1  0  0  0  2  0   57 [90.6/0.3]
tuse   0  0  0  0  1  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 145  0    2 [98.0/0.0]
milj   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 25  0    0
komm   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  6  0    0
kron   0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  3 175  0  0  0  0  1  0    4 [96.2/0.1]
och    0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  8  0  0  0    10 [80.0/0.0]
stre   0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0  0    0 [87.5/0.0]
öre    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9    1
Ins    6 97  6 11  9 14  8 17 20 10  0  2  0  0  0  0  0  0  1  1  0  1  1  0  0  0  1  0  0  3  0  0 11  3  0  2  3  0  0  2  1  2  0  9  0  2  0  0  2  0
============================================================================
```

Figure E.5. Confusion matrix: triphones, cross-word context expansion, old lexicon, eight Gaussian mixture terms

```
--------------------------- Confusion Matrix ---------------------------
      n  e  t  t  f  f  s  s  s  å  n  t  e  t  t  f  f  s  s  å  n  t  t  f  f  s  s  a  n  h  t  m  b  d  i  j  k  k  m  n  n  o  s  s  v  v  ä  ö
      o  t  v  r  y  e  e  j  t  i  i  l  o  r  y  e  e  j  t  i  j  r  j  e  e  j  r  i  u  u  i  i  e  n  a  o  r  e  e  u  c  t  t  a  e  r  r
      l  t  å  e  r  m  x  u  t  o  o  v  l  e  r  m  x  u  t  j  r  t  t  t  t  o  t  d  e  j  d  e  m  o  n  j  m  h  r  å  d  t  e
      l           a           a              v  t  t  t  t  t  t  i  t  g  t  r  t  t  t  o  t  n  o  r  n  o  e           a  o        e  c        r
                                             t  i  i  i  t  o  i  o  t  t  o  o  t  n  o  r  n  o  e           a  o        e  c                  Del [ %c / %e]
noll 1026  2  2  1  1  1  0  2  0  7  1  1  1  2  0  0  0  0  0  0  0  1  1  0  2  1  1  1  3  0  0  0  0  0  0  0  0  2  0  0  1  1  0  0  0  2  0   31 [96.5/0.3]
 ett   0 802  4 20  2  1  2  1  5  2  1  0  0 17  0  2  1  1  5  5  0  4  1  0  0  0  0  2  0  0  0  1  2  0  0  0  0  0  0  0  0  0  0  0  1  0   49 [90.9/0.7]
 två   5  6 803  0  0  4  2  1  2  0  4  0  2  2  0  0  0  0  0  1  0  3  0  1  1  1  1  1  0  1  0  0  0  0  2  0  1  0  0  0  0  1  1  0  0  0   43 [94.9/0.4]
 tre   0 15  0 719 11  2  6  5  1  6 12  0  0  7  1  7  5  1  1  3  1  0  0  0  1  0  0  1  1  1  0  0  1  0  0  0  1  0  2  0  0  0  0  0  0  0   62 [88.7/0.8]
fyra   0  1  0  3 788  1  0  0  1  4  3  0  0  0  1  1  0  0  0  0  0  1  0  0  0  0  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0   30 [97.5/0.2]
 fem   0 38  2  7  3 686  0  0  3  0  0  0  0  0  0 17  0  0  0  1  1  1 12  2  1  0  0  1  9  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0   47 [87.4/0.9]
 sex   0  3  0  5  0  0 710  0  0  0  0  0  1  0  0  0 25  0  0  2  0  0  0  0  6  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0   47 [94.0/0.4]
 sju   3  4  1  8  5  1  1 659  0  0  1  0  2  0  0  0  0  3  0  1  6  1  0  0  0  2  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0   48 [94.3/0.4]
åtta   0  1  2  0  1  0  0  680  0  0  0  0  1  0  0  0  1  4  0  0  2  0  0  1  5  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0   34 [97.3/0.2]
 nio   2 13  1 11 16  2  0  2  0 602  8  1  0  0  0  1  1  0  0 11  1  0  2  0  0  0  1  2  1  2  3  0  2  0  1  0  1  0  1  0  1  0  0  0  1  0  1   38 [87.1/0.8]
 tio   0  5  0  3  1  0  0  0  0  3 46  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0    2 [76.7/0.1]
elva   0  0  0  0  0  1  0  0  0  0  0 63  0  0  0  0  0  0  0  0  2  0  0  1  0  0  0  0  0  0    1 [94.0/0.0]
tolv   0  1  1  0  0  0  0  0  0  0  0  0 44  0  0  0  0  0  0  0  1  0  0  1  1  0  0  0  0  0  0  0  1    1 [88.0/0.0]
tret   0  1  0  1  0  0  2  0  0  1  1  0  0 138  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    6 [93.9/0.1]
fyrt   0  0  0  0  0  0  0  0  0  0  0  0  0  3 96  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0    4 [94.1/0.1]
femt   0  0  0  3  0  1  0  0  0  1  0  0  0  3  2 126  1  1  0  0  0  0  0  0  1  0  0  0  0    3 [90.6/0.1]
sext   0  0  0  1  0  0  8  0  0  0  0  0  0  1  0  0 126  0  0  0  0  0  0  0  0  0  0    3 [92.6/0.1]
sjut   0  0  0  0  0  0  0  4  0  0  0  0  0  1  3  0  0 147  3  2  0  0  0  0  0  0    4 [91.9/0.1]
åtti   0  1  0  1  0  0  0  1  6  0  0  0  0  0  1  1  0  2 159  0  0  0  0  0  1  0  0  0  1  0  0  0  1  0   12 [90.9/0.1]
nitt   0  1  0  2  0  0  1  0  0  0  0  0  1  0  1  2  0  1 172  0  0  0  0  0  0  1  0  0  1  0   10 [94.0/0.1]
tjug   1  0  0  1  0  0  1  2  0  0  0  0  0  0  2  0  0  0  0  0 159  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0   16 [94.6/0.1]
tret   0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0 34  0  0  0  1  0  0  0  0    0 [91.9/0.0]
fjor   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0    0
femt   0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0 42  0  0  0  0    1 [97.7/0.0]
sext   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 42  0  0  0    2
sjut   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 30  0  0  0    0 [96.8/0.0]
arto   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 22  0  0    0
nitt   0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  1  0 30  0  0  0    1 [88.2/0.0]
hund   8  2  1  0  6  2  0  1  2  3  6  2  5  0  0  1  1  0  0  2  2  3  0  0  5  1  3  0 333  0  0  0  0  0  2  2  0  0  0   46 [84.7/0.5]
tuse   0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0 143  0  0    3 [97.3/0.0]
milj   0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 24  0  0    0 [96.0/0.0]
komm   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  6  0  0    0
kron   0  0  0  0  1  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0 179  0  0  2    2 [97.3/0.0]
 och   0  0  0  0  0  1  0  0  1  0  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0    8 [58.3/0.0]
stre   0  0  0  2  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  0  0    0 [62.5/0.0]
 öre   0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  8    1 [88.9/0.0]
 Ins   4 203 34 47  7 13  7 13 25 20  1  1  0  6  0  1  1  2  8  1  2  0  1  0  0  0  0  0  2  3  1  0  2  1  1  0  1  1  3  0  3  2  0  0  0  2  0
========================================================================
```

```
--------------------------- Confusion Matrix ------------------------------
      n  e  t  t  t  f  f  s  s  s  å  n  t  e  t  t  f  f  s  s  å  n  t  t  f  f  s  s  a  n  h  t  m  d  e  h  i  j  k  k  k  n  o  s  s  s  ä  ö
      o  t  v  r  y  e  e  j  t  i  i  l  o  r  y  e  e  j  t  i  j  r  j  e  e  j  r  i  u  u  i  e  l  a  n  a  o  r  r  e  c  o  t  t  r  r
      l  t  å  e  m  x  u  t  o  o  v  l  e  r  m  x  u  t  t  u  e  o  m  x  u  t  t  n  s  l  t  l  r  t  g  m  o  o  j  h  m  r  å        e
      l                 a           a                 a  v  t  t  t  t  t  t  i  t  g  t  r  t  t  t  o  t  d  e  j     e        e     m  n  n              e  r
                                             t  i  i  i  t  o  i  o  t  t  o  o  t  n  o  r  n  o     r                 a  a  o                 c
                                                                                                                                          Del [ %c / %e]
noll 1042 0  2  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  2  0  0  2  1  0  3  1  1  0  0  0  0  0  1  0  1  0  0  6  1  1  0  0  0  2  0   25 [97.5/0.2]
 ett  0 806 4  18 2  7  1  1  3  0  0  0  0  9  1  5  1  0  1  2  0  6  0  0  0  0  1  2  1  0  1  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0   55 [92.0/0.6]
 två  3  0 804 2  2  6  0  1  2  0  1  0  3  0  0  2  0  0  1  0  1  1  0  1  1  2  0  0  0  1  0  0  0  0  0  0  0  3  0  1  0  0  7  0  0   44 [95.1/0.4]
 tre  0  10 1 753 13 4  2  5  1  3  5  0  0  3  0  5  4  0  0  1  0  0  0  0  1  1  0  0  1  4  0  2  1  0  0  0  0  0  2  0  0  2  0  1  0   48 [91.3/0.7]
fyra  0  1  0  5 793 1  0  1  2  1  3  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  0  1  2  0  0  0  0  0  0  0  0  0  0  0  5  0  0  1  0   25 [97.5/0.2]
 fem  0  12 2  2  3 712 0  0  2  0  0  0  0  0  0  0  21 0  0  0  0  0  1  8  2  1  0  0  1  7  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0   57 [91.9/0.6]
 sex  0  3  0  1  0  0 731 1  0  0  0  0  0  0  0  1  12 0  0  0  0  0  0  5  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0   45 [96.6/0.2]
 sju  4  0  2  6  4  1  3 662 0  0  0  0  0  4  0  0  0  0  1  1  0  6  0  0  0  0  1  0  2  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   45 [94.3/0.4]
åtta  0  1  0  2  1  1  0  0 690 0  0  0  0  1  0  0  0  1  5  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0   27 [97.7/0.1]
 nio  3  2  0  13 19 3  0  3  0 604 1  0  1  0  0  3  1  0  1  7  0  0  0  0  0  0  1  3  4  6  2  0  1  0  0  0  0  1  2  0  0  0  0  1  1   46 [88.4/0.7]
 tio  0  2  0  4  5  0  0  0  0  2  44 0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    4 [75.9/0.1]
elva  0  0  0  0  0  1  0  0  0  0  0  62 0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    2 [93.9/0.0]
tolv  0  1  1  0  0  0  0  0  0  0  0  0  45 0  0  0  0  0  0  0  0  1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0    0 [88.2/0.1]
tret  0  3  0  2  0  0  0  0  0  0  0  0  0 138 1  0  1  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    6 [93.9/0.1]
fyrt  0  0  0  0  0  0  1  0  0  0  0  0  0  1  94 1  0  6  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    3 [91.3/0.1]
femt  0  0  1  3  0  0  0  0  0  1  0  0  0  0  1 128 1  0  0  0  0  2  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    4 [92.8/0.1]
sext  0  0  0  2  0  0  8  0  0  1  0  0  0  0  0  0 125 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    3 [91.9/0.1]
sjut  0  1  0  0  0  0  0  3  0  0  0  0  0  2  4  0  0 148 0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    4 [92.5/0.1]
åtti  1  2  0  7  0  0  0  0  4  3  0  0  2  4  0  0  15 126 2  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  4  0  1  0  0  0  0  1  0   14 [72.8/0.4]
nitt  0  0  0  5  0  0  2  0  0  1  1  0  0  2  0  0  2  1  0 172 0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0    5 [91.5/0.1]
tjug  2  0  0  0  0  0  0  3  0  0  0  0  0  2  1  2  2  1  0  0 147 0  0  0  0  0  0  0  0  3  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0   19 [89.1/0.2]
tret  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  35 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    0 [94.6/0.0]
fjor  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  40 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    0
femt  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  41 0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    1 [95.3/0.0]
sext  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  42 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    2
sjut  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  29 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    0 [93.5/0.0]
arto  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  21 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    0 [95.5/0.0]
nitt  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  1  0  30 0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0    1 [88.2/0.0]
hund  8  2  0  1  6  4  0  2  2  1  1  1  6  0  0  2  1  0  0  0  2  3  0  1  7  2  1  0 328 0  0  1  0  0  0  0  3  0  0  1  0  0  0  0  0   53 [85.0/0.5]
tuse  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0 146 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    2 [98.6/0.0]
milj  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  24 0  0  0  0  0  0  0  1  0  0  0  0  0  0    0 [96.0/0.0]
komm  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  0  0    0
kron  0  0  0  0  2  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  1 176 0  0  0  0  0  0  0  0  1  0    3 [96.2/0.1]
 och  0  0  0  0  0  3  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0  0  0  0  0  0  0  0  0    9 [63.6/0.0]
stre  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  8  0  0  0    0
 öre  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  8    1 [88.9/0.1]
 Ins  4  95 9  22 9  16 6  20 18 20 2  0  1  1  0  1  1  0  2  1  1  0  0  0  1  0  1  0  4  3  0  8  0  0  2  1  0  0  2  0  9  0  2  0  1  0
==========================================================================
```