Aalto University
School of Science and Technology
Faculty of Electronics, Communications and Automation

Reima Karhila

Cross-lingual acoustic model adaptation for speaker-independent speech recognition

Master's Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Technology.

Helsinki, April 12, 2010

Supervisor: Professor Paavo Alku

Instructor: Docent Mikko Kurimo

Aalto University
School of Science and Technology

ABSTRACT OF THE
MASTER'S THESIS

Author:	Reima Karhila
Name of the Thesis:	Cross-lingual acoustic model adaptation for speaker-independent speech recognition
Date:	April 12, 2010 **Number of pages:** 12+124
Department:	Faculty of Electronics, Communications and Automation
Professorship:	S-89 Acoustics and Audio Signal Processing
Supervisor:	Prof. Paavo Alku
Instructor:	Docent Mikko Kurimo

For good quality speech recognition, the ability of the recognition system to adapt itself to each speaker's voice and speaking style is more than necessary. Most of speech recognition systems are developed for very specific purposes for a linguistically homogenous group. However, as user groups are formed out of people from differing linguistic backgrounds, there is an ever-growing demand for efficient multi-lingual speech technology that takes into account not only varying dialects and accents but also different languages.

This thesis investigated how the acoustic models for English and Finnish can be efficiently combined to create a multilingual speech recognition system. Also how these combined systems perform speaker adaptation within languages and across languages using data from one language to improve recognition of the same speaker speaking another language was investigated. Recognition systems were trained based on large Finnish and English corpora, and tested both on monolingual and bilingual material.

This study shows that the thresholds for safe merging of the model sets of Finnish and English are so low that the merging can hardly be motivated from the point of view of efficieny.

Also it was found out that the recognition of native Finnish can be improved with the use of English speech data from the same speaker. This only works one-way, as the foreign English recognition could not be significantly improved with the help of Finnish speech data.

Keywords: automatic speech recognition, multi-lingual acoustic modelling, acoustic model adaptation, cross-lingual speaker adaptation

Aalto-yliopisto
Teknillinen Korkeakoulu

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Reima Karhila	
Työn nimi:	Akustisten mallien adaptointi kielten yli puhujariippumattomassa puheentunnistuksessa	
Päivämäärä:	12.4.2010	**Sivuja:** 12+124
Osasto:	Elektroniikan, tietoliikenteen ja automaation tiedekunta	
Professuuri:	S-89 Akustiikka ja äänenkäsittelytekniikka	
Työn valvoja:	Prof. Paavo Alku	
Työn ohjaaja:	Dosentti Mikko Kurimo	

Laadukas puheentunnistus vaatii tunnistussysteemiltä kykyä mukautua puhujan ääneen ja puhetapaan. Suurin osa puheentunnistusjärjestelmistä on rakennettu kielellisesti yhtenäisten ryhmien käyttöön. Kun erilaisista kielellisistä taustoista tulevat ihmiset muodostavat enemmän ja enemmän käyttäjäryhmiä, tarve lisääntyy tehokkaalle monikieliselle puheentunnistukselle, joka ottaa huomioon murteiden ja painotusten lisäksi myös eri kielet.

Tässä työssä tutkittiin, miten englannin ja suomen puheen akustisia malleja voidaan yhdistellä ja näin rakentaa monikielinen puheentunnistin. Työssä tutkittiin myös miten puhuja-adaptaatio toimii näissä järjestelmissä kielten sisällä ja kielirajan yli niin, että yhden kielen puhedataa käytetään adaptaatioon toisella kielellä. Puheentunnistimia rakennettiin suurilla suomen- ja englanninkielisillä puhekorpuksilla ja testattiin sekä yksi- että kaksikielisellä aineistolla.

Tulosten perusteella voidaan todeta, että englannin ja suomen akustisten mallien yhdistelemisessä turvallisen klusteroinnin raja on niin alhaalla, että yhdistely ei juurikaan kannata tunnistimen tehokkuuden parantamiseksi.

Tuloksista nähdään myös, että äidinkielenä puhutun suomen tunnistamista voitiin parantaa käyttämällä vieraana kielenä puhutun englannin dataa. Tämä mekanismi toimi vain yksisuuntaisesti: Vieraana kielenä puhutun englannin tunnistusta ei voinut parantaa äidinkielenä puhutun suomen datan avulla.

Avainsanat: Puheentunnistus, monikielinen akustinen mallinnus, akustisten mallien adaptaatio, kielten yli tapahtuva puhuja-adaptaatio

Acknowledgements

This Master's thesis was started in the Laboratory of Computer and Information Science in Helsinki University of Technology in 2008, continued in the Faculty of Information and Computer Sciences in the same university in 2009 and was finally finished in 2010 in the same department at the Aalto University School of Science and Technology.

I want to thank my colleagues here at ICS speech group and abroad. I am grateful to my instructor Mikko Kurimo for a chance to work on this thesis and my gratitude extends to all European tax-payers, as this thesis has been funded by you!

The saying "if you can't explain your research to your grandmother, you don't know your subject well enough" is attributed to Richard Feynman, and I have written the first chapters of this thesis accordingly. Explaining this work to my grandparents would at the moment require some shamanistic abilities, but I hope my living relatives and friends might understand something of what I am doing.

To avoid domestic disputes I also thank Niamh for her patience.

And finally, as a peculiar detail, I would like to mention mr Kleinberg senior, who once told about the engineers at the Helsinki shipyard; According to them, any engineer who does not deal with mechanical stress and strength calculations is a *hölpön-pölpön-insinööri*, a mumbo jumbo engineer. So now I present you, dear reader, my *hölpön-pölpön* -thesis.

Otaniemi, April 12, 2010

Contents

Abbreviations

CMLLR	Constrained MLLR
CMS	Cepstral mean substraction
GMM	Gaussian Mixture Model
HLDA	Heteroschedastic LDA
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit (software package)
IPA	Internation Phonetic Alphabet *or* International Phonetic Association
LDA	Linear Discriminant Analysis
LER	Letter Error Rate
LVCSR	Large Vocabulary Continuous Speech Recognisers
MFCC	Mel-Frequency Cepstral Component
MLLR	Maximum Likelihood Linear Regression
ML	Maximum Likelihood
PCM	Pulse Code Modulation
RT	Real-Time
SAMPA	Speech Assessment Methods Phonetic Alphabet
SD	Speaker-Dependent
SI	Speaker-Independent
STC transform	Semi-tied covariance transform
WER	Word Error Rate

List of Figures

List of Tables

Chapter 1

Introduction

Applications of speech recognition are increasing as speech recognition technology is slowly growing mature. Already telephone number enquiries, taxi ordering or simple banking operations can be made using a speech recognition system via a telephone. Some consumer electronic products like mobile phones and computers can be set up to respond to voice commands.

This thesis covers some aspects of speech recognition in multiple languages and is one in a long line of theses written at the the Faculty of Information and Computer sciences in Aalto University School of Science and Technology.

Speech recognition research at Aalto university

Starting with Jalanko, 1980, the Faculty of Information and Computer sciences has been developing Finnish speech recognition technologies. In the last years, research in large-vocabulary speech recognition has been bountiful.

One of the biggest obstacle to prevalence of speech recognition technology in user interfaces of consumer products is the wildly varying way that languages are spoken. Some people speak fast, some slowly; The tone and pitch of speech varies according to physical attributes and emotional state of the speaker. Beside personal differences, also regional differences cause problems to speech recognisers. American English speech recognition systems do not work sufficiently well on British English. Local dialects create small problems for a foreign tourists and huge problems for a speech recogniser. Even though numerous advanced techniques exist, the question how a recogniser can adapt itself to various speakers' voice qualities and speaking style is a question that is not yet fully solved.

To overcome the problems of speaker variety, Varjokallio, 2007 investigated sub-

space methods to improve the robustness of acoustic modelling and Remes, 2007 and Mansikkaniemi, 2010 investigated speaker adaptation techniques.

Speech recognisers are often quite vulnerable to background noise. Similar techniques to speaker voice adaptation are used to improve recognition in noisy conditions. Additionally, an ever-growing number of techniques are developed specifically to counter the problem of background noise.

Kallasjoki, 2009 and Keronen, 2009 developed methods to improve recognition of noisy speech, the former with a new spectral enveloping method for feature extraction and the latter with parallel model combination for noise removal.

Another restriction is the computation power required by the more complex speech recognisers. If the interaction of the user and the device is not highly restricted with the help of specific questions, as it is in a taxi or banking service, the vocabulary and grammatic knowledge of the recogniser need to be very large. Therefore proper recognisers for dictation tasks are rare and need to be custom-made for the task. An example of a complex recogniser with complex grammar is a dictation device for medical doctors, to be used to produce the minutes of a patient visit.

Creutz, 2006 investigated automatic morphological analysis of large text corpora, and Siivola, 2007 and Hirsimäki, 2009 applied these principles to improve language modelling in morphologically complex languages, leading to a better performance of large vocabulary Finnish speech recognisers.

All in all, the TKK recogniser is a formidable research tool and performs very well by international standards. However, the TKK recogniser is not used in this thesis. Instead, the HTK speech recognition toolkit (Young *et al.* , 2006) is used. This is due to the demands in compatibility and collaboration in the project that funded this thesis. The HTK toolkit does not include refined methods for noise-robust recognition and is rather clumsy with the large language models required by morph-based speech recognition. What HTK can offer is an internationally known standard platform for HMM-based recognition and optimised performance in some of the training and testing tasks.

The EMIME project

This thesis is a part of the European Union funded research project *Effective Multilingual Interaction in Mobile Environments*, EMIME.

The EMIME project is a collaborative research project between the *University of Edinburgh* (UEDIN), *University of Cambridge* (UCAM), *Aalto University school of science and technology / Helsinki University of Technology* (Aalto / TKK), *Nagoya*

Figure 1.1: The EMIME system, pictured below, changes the "colour" of the synthesised voice to match the current user. An improvement over the old systems, above, that always use the same synthesised voice. From Mainichi Newspaper, May 20th 2008.

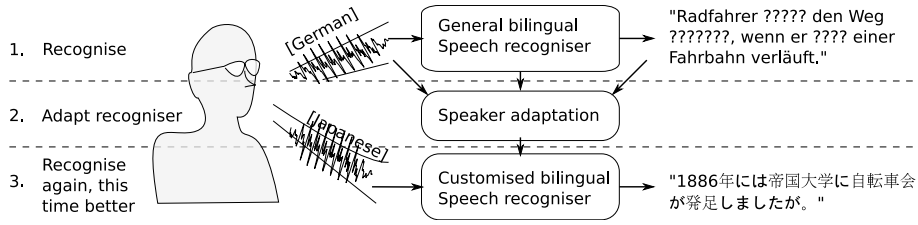


Figure 1.2: Adaptation is a technique to customise a speech recogniser for a particular speaker. When a new user first uses a speech recogniser (1), as well as recognising the speech, the system customises itself to the user(2), so that when the same user speaks again, the recognition is improved(3). The goal is to be able to do the improvement part across the language barrier.

institute of Technology (NIT), *Institut Dalle Molle d'Intelligence Artificielle Perceptive* (IDIAP) and *Nokia Corporation* Beijing office.

The goal of this 3-year project spanning from 2008 to 2011 is to develop a prototype of a hand-held device that recognises speech in several languages, translates the speech sentence by sentence and synthesises speech from the newly translated sentences in another language, mimicking the original speaker's voice. The general principle is illustrated in Figure 1.1

The prototype version of the EMIME system will be a software package for certain modern mobile phones. A client-server -architecture will be used, where the mobile device acts only as a terminal that records and outputs speech. Automatic speech recognition (ASR) requires a significant amount of computing power. By using this kind of architecture, speech recognition, translation and synthesis can be done on high-capacity servers instead of the small client.

Still, whether using a dedicated small device or a server with many users, the amount of resources available per user are always limited. Therefore all of the three complex processes of recognition, translation and synthesis of speech must be optimised as far as is possible. The main restraints are processing time and memory requirements.

Goals of this thesis

One typical technique for easing some of the computational burden of speech recognition is to combine the models of acoustically similar sounds.

I will investigate the possibility of doing this across languages by trying to find similar sounds in Finnish and English languages. Additionally, I will investigate the possibilities of using speech data across languages to improve recognition, as

illustrated in Figure 1.2.

To help understand the objective better, I will quickly introduce the reader to the basic concepts of speech and speech recognition and give a little insight in the inner workings of a speech recognition system in chapters 2 and 3. In Chapter 4 I will explain the common standards used to describe the quality of a speech recognition system. Those more familiar with the subject are invited to skip these.

In chapters 5 and 6, I continue the introduction to speech recognition by examining in more detail two particular areas of interest in speech recognition: Multilingual recognition and speaker adaptation. These are meant for the main audience of this thesis (the supervisor and the reviewing board), who are already familiar with the terminology and theory. If the reader is not familiar with the field, it is perfectly all right to not to understand too much of these.

Finally, I will present my test set-up in Chapter 7, and give a detailed explanation of my tests and results in Chapter 8. The impatient reader can skip straight to Chapter 9, where I present my conclusions.

Chapter 2

Basics of speech production and perception

Spoken language consists of uttered sentences. The sentences consist of words and the words consist of groups of "sounds". These utterances are caught by the recipient's ears and after some neural processing the message carried by the utterance is understood by the recipient.

This is everyday knowledge, and is enough for a casual conversation about communication, but a scientific approach to speech requires that the terminology used by the researchers is well defined. In this chapter I will quickly go through some essential terms and after that introduce the basic ideas on how speech is formed and how it is heard.

2.1 Terminology of utterances

The sounds that make up spoken speech are, in linguist's terms, called **phones**. Whereas in most western writing systems a written word is made of letters from left to right, the spoken word is made from phones that follow each other in time. If there is a need, as there is in this thesis, to describe a spoken word in written form, the individual phones that make up the word are written down, usually again from left to right. This is called **transcription**.

To be exact, a phone is only a realisation of a **phoneme**. A phoneme is a low level unit of spoken language. It is the smallest unit that carries meaning. This means that changing a phoneme in an utterance can change the meaning of the utterance. The phoneme is a somewhat abstract concept, as it is not seen as itself, but only by its representative phones. Just like every hand-written letter is different from other

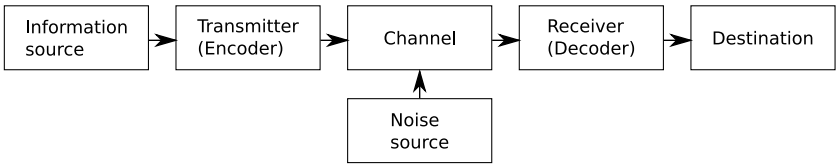


Figure 2.1: Communication channel according to Shannon, from Shannon, 2001.

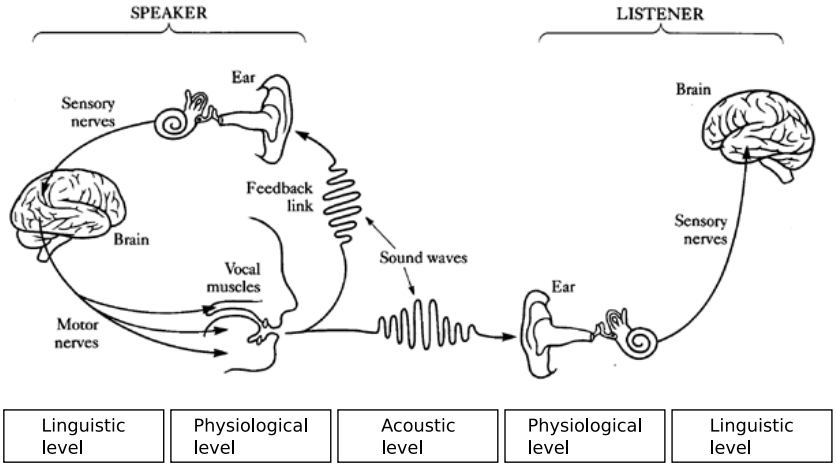


Figure 2.2: The speech chain, from Denes *et al.* , 1970.

hand-written letters, so are phones unique, but they are always representations of a phoneme like the hand-written letter is a representation of the "ideal" letter.

The way the phones are marked on paper must of course be standardised in order for linguists to exchange their ideas. *The International Phonetic Association* has produced a standard set of characters for phonetic transcriptions. This is called the International Phonetic Alphabet (IPA). In the **IPA alphabet**, phones are represented by letters that are most familiar to people who are used to the Roman alphabet. Especially in the case of phonetic languages, where the way a word is pronounced can be seen from the way it is written, care must be taken not to confuse the concepts of letters and phones. Transcriptions of phones are often, including in this text, marked by slashes on both sides, like this: /a/.

When discussing speech recognition, we mainly talk about phones. Or rather, we are talking about mapping phones into phonemes, that can be used to recognise what words are being uttered.

For the engineer, notorious for not caring about the subtleties of terminology or the finer theories and just sprinting to a practical, technical solution; questions about the true nature of speech are irrelevant. An engineer sees speech as a communication

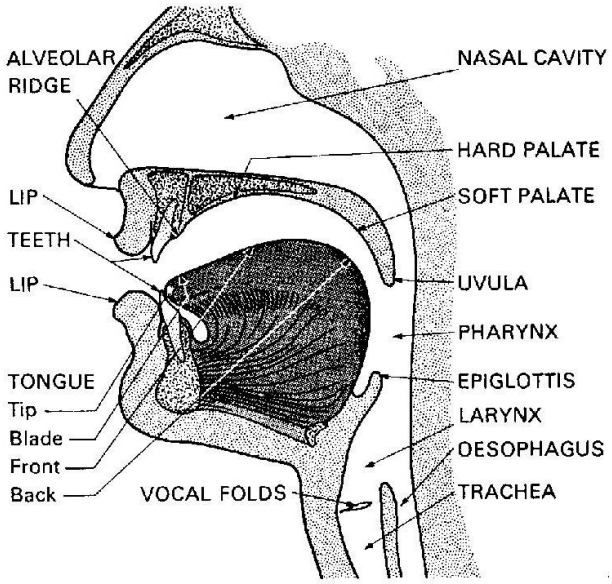


Figure 2.3: The vocal tract. From the University of Aberdeen, Linguistics and Language Department's learning resources.

system (Figure 2.1), or as a slightly less general "speech chain" (Figure 2.2), and the spoken language as a continuous analog acoustic signal that contains **information**.

The problem of recognising words in speech in the speech recognition engineer's mind is reduced to analysing the acoustic signal. This is done by matching phone models to the signal according to combinations possible in the given language.

In a typical engineering manner, the terms "phoneme" and "phone" are not used in this thesis properly. The term phone is used to cover both.

2.2 Speech formation

Let us now have a quick look at how the human body produces speech. The **linguistic level** in Figure 2.2 can be further broken down into two sub-levels: On a **semantic level**, we are interested in the meaning of what is to be communicated. On a **syntactic level** we are dealing with how to say things with the syntax of a given language. These are important questions in the fields of psychology and linguistics, but we will skip them in this work, as well as the neurological aspects of how the articulatory organs are controlled by the neurological system. We will only concentrate on speech itself, and on the physics and physiology of speech signal creation.

After the speaker has decided what to speak and which words to use, the process

on the **physiological level** deals with the creation of the speech signal. On this level, the speaker controls the articulatory system of the body.

Speech as an acoustic waveform is formed by the air flow pushed out from the **lungs** through the **vocal folds**. The fluctuation of the vocal folds is the **source** of voiced speech. When speaking, the vocal folds create a simple monotonous sound, called **the glottal excitation signal**. The excitation signal is **filtered** (i.e. some of the signal's frequencies are amplified and some are dampened) by the **vocal tract** to create more colourful sounds. The vocal tract consists of parts of the throat and the mouth, and sometimes the nasal cavity, as shown in Figure 2.3.

The basic sound of **Vowels** like /a/ or /e/ is produced by the vocal folds rapidly opening and closing rhythmically while air is pushed through them. This opening and closing causes the air to come out from the lungs in quick, forceful bursts. The speed of opening and closing is typically around 80-130 times per second (Hz), which in music terms would mean around 4800-7800 beats per minute. The single impulses making the sound cannot be heard - instead we hear a tone.

The different sounds of different vowels are formed when the air pulses coming from vocal cords push through the mouth and lips (and sometimes the nasal cavity) unobstructed but altered by different "bottlenecks" formed by the position of the tongue, lips and other parts of the vocal tract. The reader is invited to try out how moving the tongue and lips changes the sound of a vowel. This is easy to try, since it is easy to continue pronouncing a single vowel.

As opposed to vowels, forming **consonants** requires that the vocal tract is blocked. There are many ways to do this and sometimes the results resemble vowels.

Fricatives are formed when the air flow is obstructed so lightly, that the turbulence of the air flow creates "hissing" sounds like /s/. One of the subgroups of fricatives are **tremulants**, where turbulence creates a rhythmic tapping of some part of the vocal tract creating sounds like /R/. **Nasals** like /m/ and /n/ use the nasal cavity to form "humming" sounds when the lips stop air from escaping through the mouth. In **approximants** like /l/ the flow of the air is not really blocked - there remains a very small passage where the sound is formed.

Stop consonants or **plosives** are formed by the vocal cords closing, building pressure and releasing it all of a sudden. Because this process requires the build-up and a sudden burst, it is not possible to keep on pronouncing a clusil like /k/ or /t/. Figure 2.4 shows the utterance "koko" by a Finnish female speaker. Here the plosive /k/'s show as almost blanks, whereas the vowel /o/-vowels create a waveform with a comparatively high amplitude.

To sum up the previous paragraphs: air coming from the lungs through the vocal

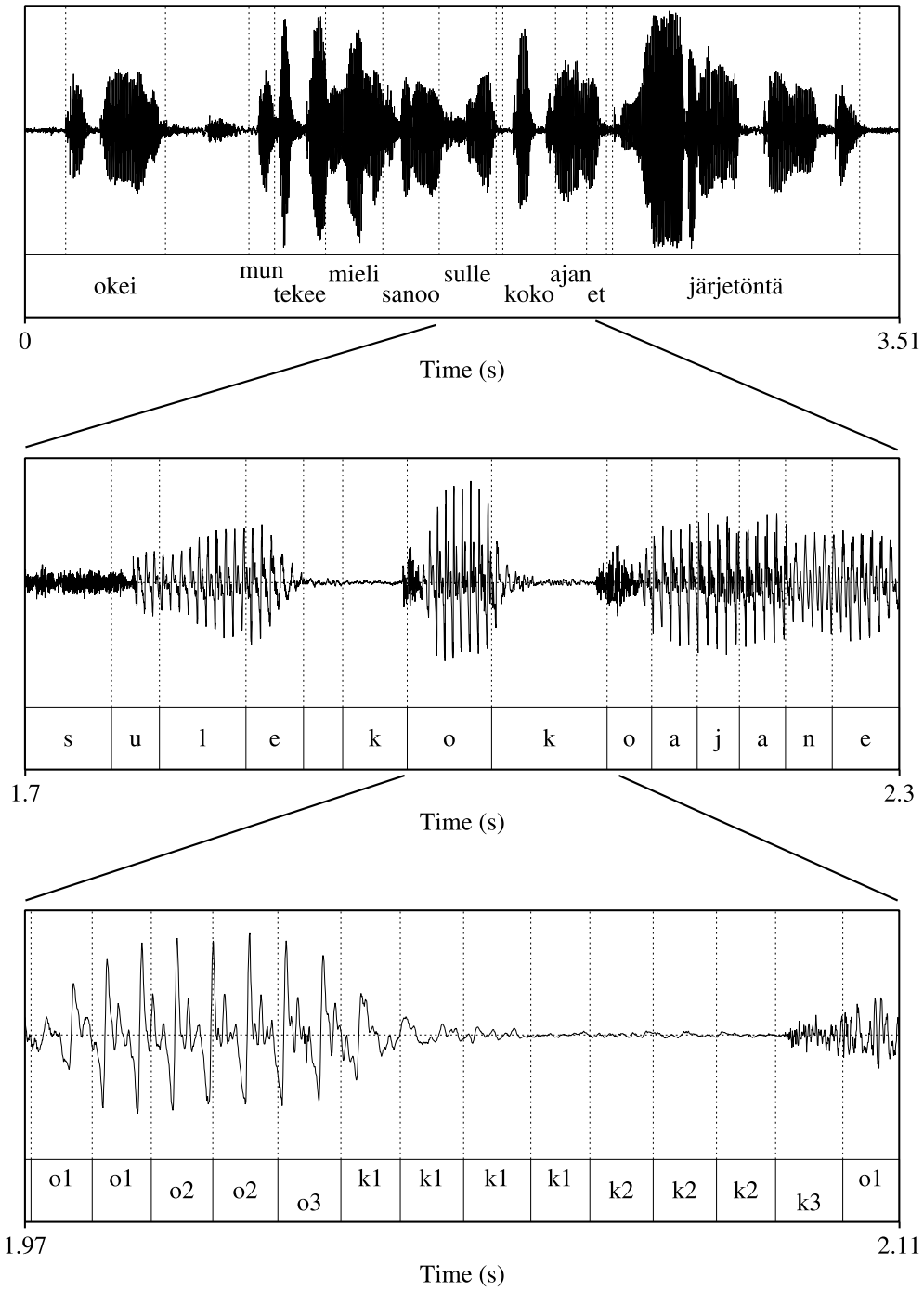


Figure 2.4: Waveform of utterance by a Finnish female speaker on various levels of detail. On the top there is the waveform of the whole sentence. Looking at a few words (middle), it is easy to see how hardly any boundaries exist between words in spoken language: words are glued together almost seamlessly. On the bottom there are waveforms for a typical vowel (/o/) with its regular, repetitive pattern and a clusil (/k/) with its short break before a noise burst. The bottom image also shows phones analysed in 10ms units and segmented into a beginning, middle and end part. This is explained in section 3.6.1.

cords makes explosive sounds, like: BOOM BOOM BOOM, but so fast that if we were to hear it unaltered, we would hear a simple tone. This sound is altered by our tongue, lips, nasal cavity etc. to produce most of the different sounds we make when we speak. We engineers like to call this alteration process **filtering**. In the case of stop consonants, the vocal tracts close for a while. Therefore there is a small silence before a few bigger booms and then the BOOM BOOM BOOM pulse train continues again.

2.3 Speech waveform and the ear

On the physiological level, the listener is much more passive than the speaker. Using the ears does not require active effort. The earlobe directs the sound waveform onto the tympanic membrane (better known as the eardrum) that directs the energy of the signal further to the inner ear.

As I have described above, speech consists mostly of the pulse train that is shaped by the vocal tract. As a result, if we were to investigate the acoustic signal as a waveform, we notice regular repetitive patterns. In Figure 2.4, a very good example of a regular pattern can be seen on the bottom image, during the pronunciation of the vowel /o/, on the left side of the waveform diagram.

Knowing that this pattern is from around 8-13 milliseconds (ms) long depending on the speaker's vocal tract's physical properties, and that a majority of phones have a duration between 50 ms and 150 ms (Pols *et al.* , 1996), we can see that the acoustic wave pattern is repeated a good few times during the pronunciation of a single vowel.

This means that the signal can be broken down into frequency components. This is exactly what the ear does. Inside the ear, after the eardrum and its magnificent bones Malleus, Incus and Stapes, there is a narrowing tube called the **cochlea**. The cochlea is a relatively long tube, narrowing towards its end, and curled into a spiral. It can be seen on the right-hand side in the figure 2.5.

The resonations caused by the acoustic signal on the eardrum is transmitted to the cochlea by the aforementioned bones. The cochlea is filled with a jelly-like substance that resonates in tune with the rhythm of the acoustic signal. Depending on the frequencies of the acoustic signal, the resonation will be stronger in some parts of the cochlea. The cochlea is lined with small, hair-like *neural receptor* cells, that are activated when they are moved - hence the resonation caused by the acoustic signal will cause a sensation of hearing a certain frequency. If the resonation is too powerful, the neural receptors can be damaged. Damaged receptors can temporarily

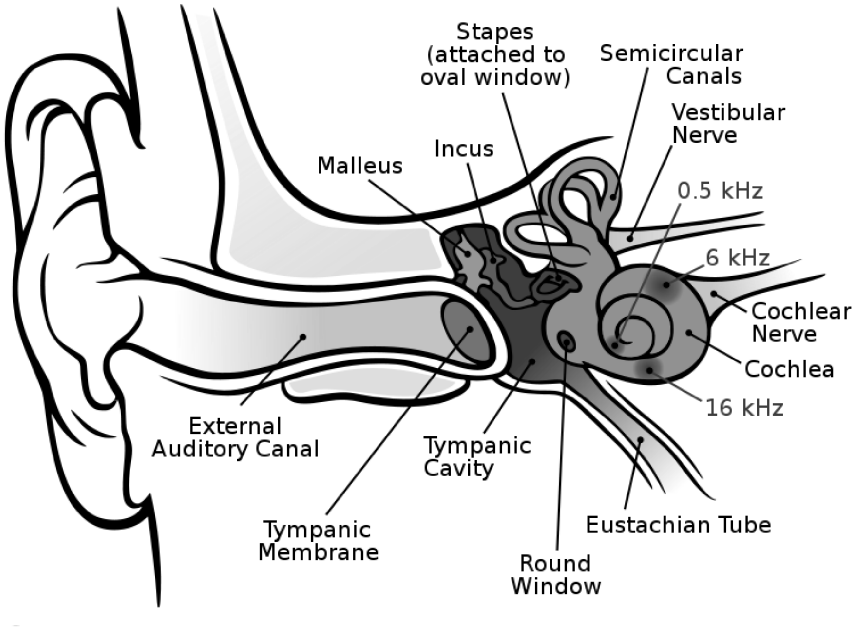


Figure 2.5: Diagram of the ear. Some vibration points in the cochlea are shown. By Chittka & Brockmann, 2005, available from Wikimedia commons.

or permanently raise the hearing threshold and can lead to Tinnitus.

Now we step on to the linguistic level of the recipient. The neural receptors are connected to the vast neural network of the brain. The nerves collect receptor information from different frequencies and combine and process this information to create the sensation of hearing and ultimately the understanding of the content of the speech. How this is done is outside the scope of this thesis, but the interested reader can read more about this in Goldstein, 2001.

The hearing system has a lot of redundancy. A lot can be left out of the acoustic signal and still the listener can understand the original sentence. This is mostly due to the neural processing, and is investigated by the field of *psychoacoustics*. Knowledge on what aspects of the signal are most important for the hearing sensation can be used as a guideline for manipulating speech signal before the actual recognition process. The parts of the signal that are important for hearing and understanding can be emphasised. The less important parts can be left out of the recognition process in order to reduce the complexity of the speech recogniser and thus speed up the process.

Chapter 3

Speech recognition for the uninitiated

In this chapter we will go through the basic principles of speech recognition and try to describe the various parts of a speech recognition system. Mathematical discussion is avoided as much as possible, but a background in physics and statistics might be required to understand the various models.

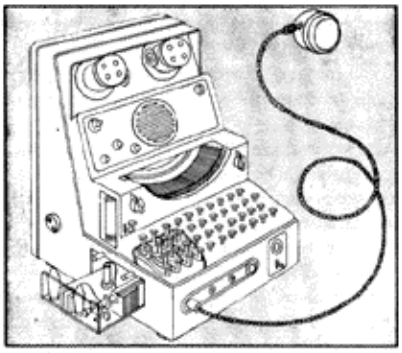
3.1 What is speech recognition

If you ask an engineer what a speech recogniser is, the answer might be something like this: The term "speech recogniser" could be applied to a system that takes speech as input and gives corresponding text strings as output, whether straight to the user or to the next software component. A straightforward example of the previous type is a **speech-to-text** system, that transforms spoken speech into written text output, as shown in Figure 3.1

In the words of a a non-engineer, this would be an electronic secretary, that performs dictation tasks.

As part of a larger system, a speech recogniser is a small input component in the device. Consider the devices in Figure 3.2: Speech recognition is only a very small part of far more complex systems that bind meaning to the recognised words and react accordingly.

For the purposes of this thesis, I will use the term 'speech recogniser' only in reference to a speech-to-text system.



Supersecretary of the coming age, the machine contemplated here would take dictation, type it automatically and even talk back if the author wanted to review what he had just said. It is somewhat similar to the Voder seen at the New York World's Fair. Like all machines suggested by the diagrams in this article, it is not yet in existence (*LIFE 19*(11), p. 114).

Figure 3.1: Dictation machine as imagined decades ago. It is surprisingly close to modern, realised speech recognition dictation systems, where a desktop computer creates transcriptions of speech recorded by a mobile recorder, phone or local microphone. From "adventures in cybersound" by Marc Berrnier (http://www.acmi.net.au/AIC/BUSH_BERRNIER.html).

Figure 3.2: Perfectly working speech technology is common in science fiction popular culture. *The T-800 series terminator* (left) and *Robby the robot* (right side of middle photo) are fictitious autonomous devices that observe their social surroundings with seemingly perfect speech recognition systems, and can also communicate with humans by speech synthesis. A piece of science-fiction has already become reality: Sony's *Aibo* robot dog (right) understands 100 voice commands. (Pictures from the films Terminator 2 (Carolco Pictures), Forbidden Planet (Metro-Goldwyn-Mayer) and Sony Europe's website.)

3.2 Speech recognition as pattern recognition

What we call pattern recognition should in the case of computer programs be called classification. Classification is basically just the task of sorting observations into predefined boxes. These boxes are often called classes. As a concept, this is not rocket science. Some of the formulas might scare the less mathematically enthusiastic readers, but the reader can be assured that the principle is simple, even when the mechanism is not.

The ability of humans to recognise various objects with incomplete information is based largely on context and previous experience. Similarly, for a computer program to recognise an object, the computer needs prior information about it. **Machine learning** techniques are used in the shaping of the boxes. Machine learning techniques do have their limitations – They can be effective in refining a system, but until the unforeseeable future, human initiative and careful planning is required to build the basis of the system.

A probabilistic speech recogniser calculates the probability that any one observation belons to one of the different classes known to the system. The observation is then assigned to the most probable class. The calculation requires heavy preprocessing of the data, which is explained in more detail in section 7.5. The sequence of classes then gives the recognition result.

3.3 Structure of a typical speech recogniser

A typical speech recognitions system consists of a **decoder**, that matches **acoustic models** onto a preprocessed **input signal**, according to a **language model**, as shown in Figure 3.3. We'll go through the parts of the speech recogniser in as gentle and non-technical manner as possible.

3.4 Preprocessing

The aim of signal preprocessing is to bring the recognisable data into a form that can be easily evaluated by the computer and to compress the data as much as possible to bring down the computational cost of analysing it.

The speech recognition process is an example of classification task. In order to classify properly, we need to define the objects of classification. We will use uniform length segments of the speech signal. Classifying waveforms would be problematic, because the waveform shape is very different depending on the sample start time.

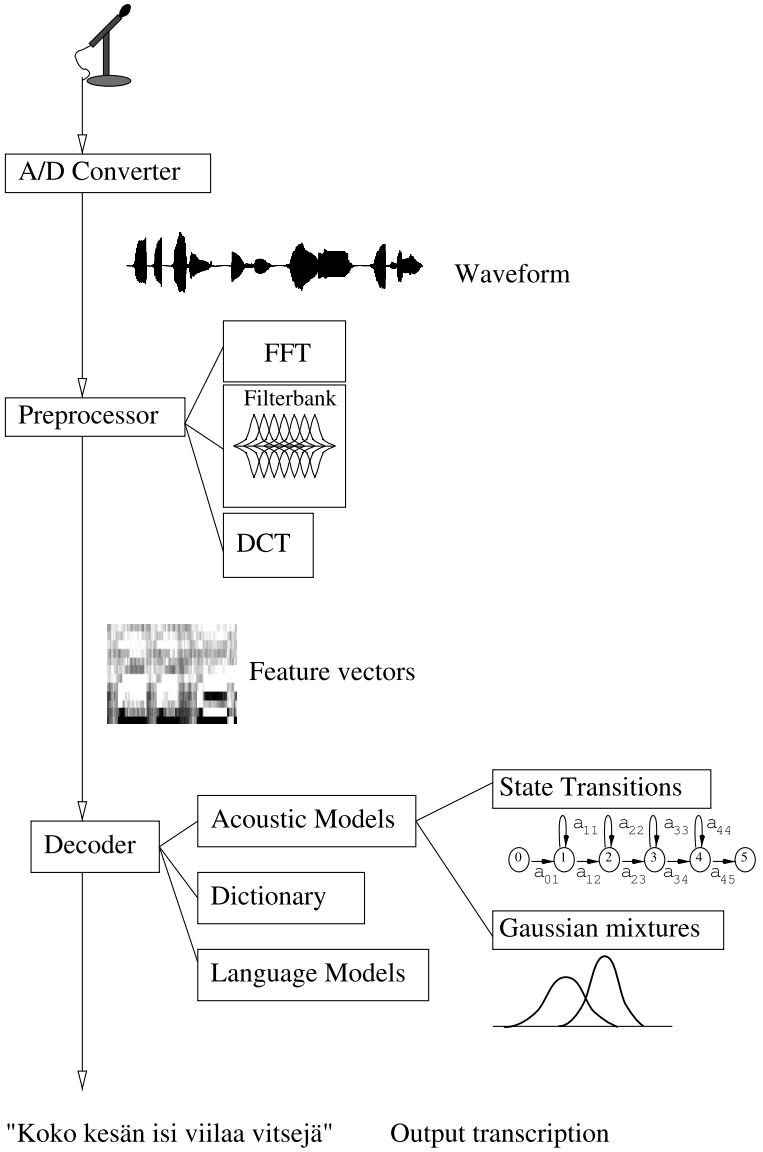


Figure 3.3: A block diagram of a typical HMM-based speech recogniser.

So we will first transfer the signal to frequency domain, where the short-time signal will have a similar form no matter where the sampling started. This transform is usually done with a Fourier transform.

We will then use further mathematical operations to reduce the amount of redundant signal data. We will reduce the amount of data radically by using Mel-scaled filter banks - these attempt to model the current ideas about the sensitivity thresholds for different frequencies in the human ear - and compress this even more with a cosine-transform, that will reduce the correlation between the samples. The cosine-transformed spectrum is usually called cepstrum, and the final samples are Mel-Frequency Cepstral Coefficients (MFCC). Other preprocessing methods exist, the most popular being Perceptual Linear Prediction (PLP) analysis. An endless amount of variations for these two feature extraction processes also exist.

As a result of preprocessing the utterance to be recognised is converted into a stack of vectors, which consist of a dozen or so coefficient values. To avoid missing any essential information, the samples overlap each other. In this work, there is one vector for each 10ms of acoustic data, and each vector covers 25ms. This means that all samples will be shared by several vectors.

With this 10ms timestep, one second of speech is represented with 100 vectors of 13 dimensions and thus by 1300 numerical values.

When running the recogniser, also the change rate of features is modelled, and the change rate of the change. So the models are built to recognise 13 feature values, 13 delta values and 13 delta-delta values giving a total of 39 dimensions for each feature vector.

3.5 Decoding and recognition tasks

The heart of the speech recognition system is the software component that calculates and keeps track of probabilities of different recognition hypotheses for the data to be recognised. This is the decoder.

3.5.1 Token-pass decoder

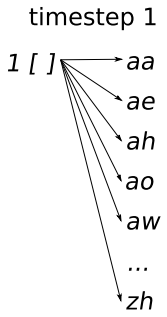
The decoder goes through every part of the speech segment and calculates the probability that given sample belonging to each of the possible models. The possible phone models and phone combinations are listed in the recogniser's **lexicon**,. The lexicon is a dictionary that has all the words that the recogniser “knows” and their associated phonetic transcriptions i.e. pronunciations.

Theoretically this would be everything needed for recognition: acoustic models, a lexicon and some data to be recognised. This however quickly leads to either quality or computational difficulties.

A token-pass decoder constrains the search-space via the use of tokens to keep track of possible recognition hypotheses. This is illustrated by a simplified example, where phone duration is not taken into account.

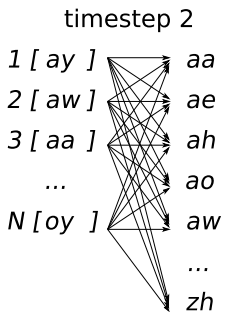
The recognition starts at the beginning of the data and proceeds one time-step at the time. As it goes through the recognition data, the decoder keeps in the computer's memory a list of most probable paths. At each time-step, every hypothesis in the recognition hypothesis stack emits tokens to all possible models that might continue the hypothesis.

At first, the hypothesis stack is empty and the list of possible paths is a list of all models of phonemes that can be at the start of a word. Tokens are then emitted to all these models:



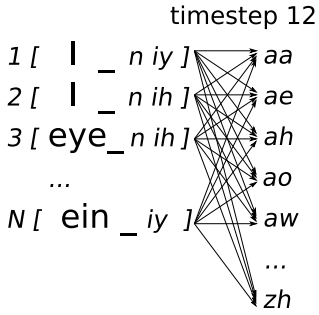
Once the tokens are passed, the probability of the acoustic features of that time-step belonging to the relevant model is calculated and added to the probability count of the path (not shown). Then the most probable hypotheses are kept and others are dropped to conserve memory and processing time.

So, the best N hypotheses are saved - shown in brackets here - and then every hypothesis passes tokens to all possible models:



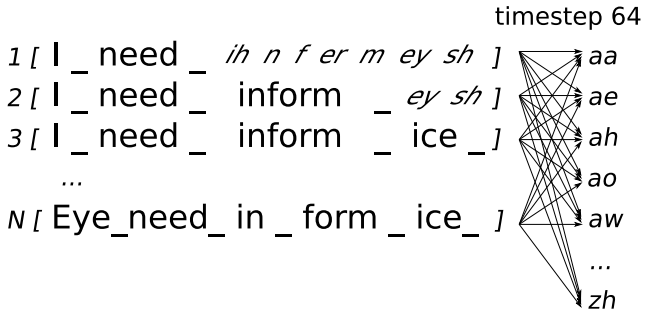
If a path has a phone sequence corresponding to a possible word, like the words

"I", "eye" and "ein" in the following diagram, the word is saved into the path and the recognition of models continues:



As the recognition progresses, the tokens carry longer and longer paths. At some points, the paths of the remaining tokens might merge and more memory can be saved. The token-pass algorithm is intended to have control on the search space size - and thus on memory consumption and required computation time.

We will shortly see why the decoder places a higher probability on "I" rather tha "eye" for the first word.



When the decoder reaches the end of the recognition data, the most probable path is taken as the strongest hypothesis and thus the end result of the recognition process.

The main constraining parameters are the maximum hypothesis stack size, and the **beam width**. If the probability of a path falls below the probability of the most probable hypothesis by more than the beam width, then that path is discarded from the stack.

A very important part of decoding is to have an efficient method of constraining the search space for possible models. Typical decoders like the HTK or TKK decoders (Pylkkönen, 2005) create a hierarchy of model networks, with a HMM subphone models at the bottom, a phone network at the middle and a word network at the top.

The network should always be created according to the task the recogniser has

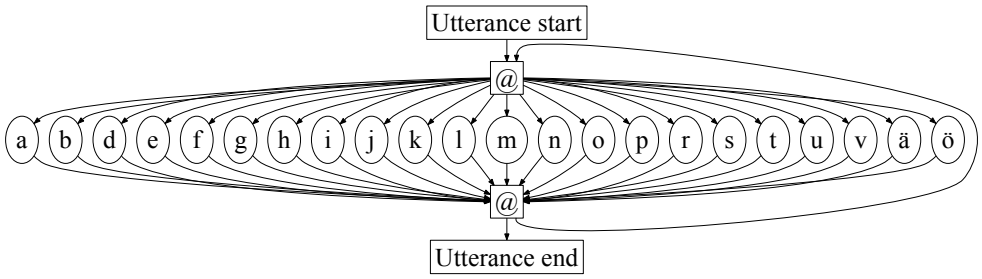


Figure 3.4: An example of a recognition network used by a Finnish phone recogniser. The @-boxes represent empty nodes, that are used only to clarify the network diagram. Without these null nodes, there would have to be arrows from every letter going back to every other letter.

to perform. We will now have a quick look at different recognition tasks and their associated higher-level networks.

3.5.2 Phone recognition

A very basic task is **phone recognition**. The constricting framework is a list of possible phones, like in Figure 3.4. The recogniser matches each part of the spoken speech signal to a predefined list of phones and looks for the best match. No word-level network is needed.

With languages like Finnish, where there is a strong correlation between written letters and spoken phones, we could recognise a sentence as a sequence of phones, by allowing sequences of phones as in the recognition. We could then hopefully be able to infer what word could be hidden in the resulting output phone sequence. For example, the utterance

se on aika raskas proseduuri että

recognised using a simple phone recogniser gives us:

tseoavaraskasb posedurötä,

which, at least in the eye of a Finnish-speaker, has a slight resemblance to the original utterance, but does not really make sense.

3.5.3 Isolated word recognition

Another basic task is **isolated word recognition**. The recogniser matches the spoken speech signal to a predefined list of words and looks for the best match. This

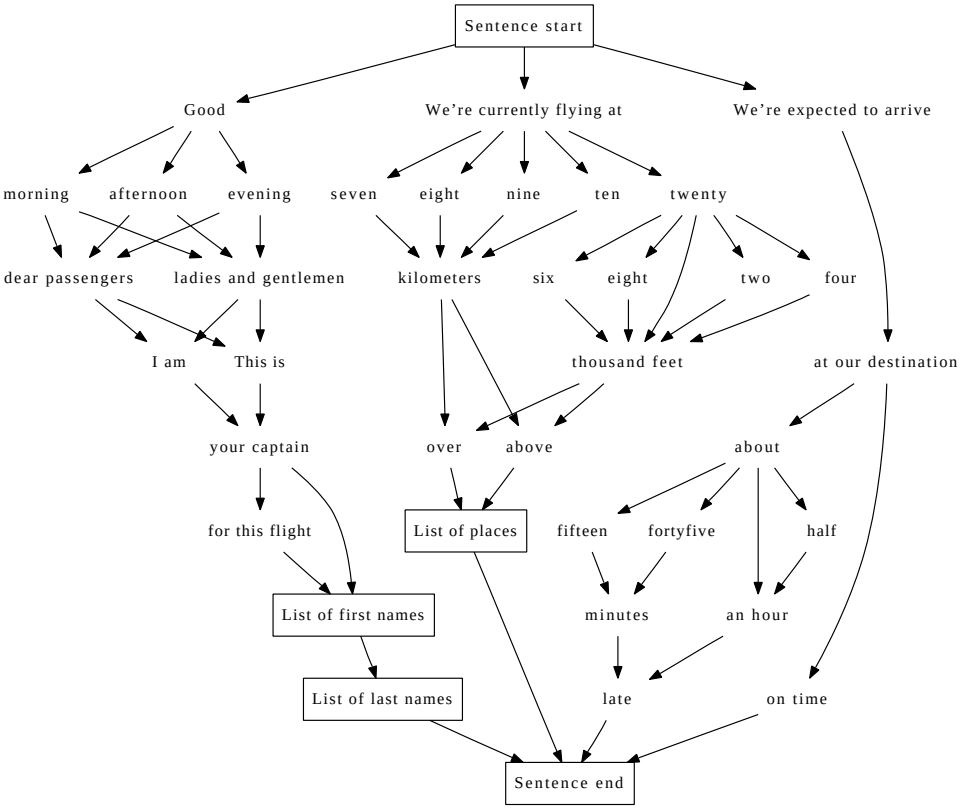


Figure 3.5: An example of finite-state grammar network.

requires a **lexicon** (pronunciation dictionary) which is a list of mappings between letters and phones that tells which phones appear after each other in a given word.

An application could be a telephone service, where the user is asked for short and specific input from a predefined list. The list of possible choices might be too numerous to be listed out by the service, but nevertheless the choices are limited in a way that is understood by the user.

For example, a ASR bus timetable service could ask the user to speak the desired bus stop name or bus line number and then tell the user when the next avaialable bus comes.

3.5.4 Finite-state grammar speech recognition

The captain of an aeroplane never seems to have a decent microphone, but still the passengers somehow make sense of what the obligatory mid-flight speech is about. Using the same principle, the most prevalent task for ASR systems deployed to consumer use is **finite-state grammar speech recognition**, where the recognition

is done with a strictly limited network of possible word combinations. This grammar is a finite-state **network** that represents rules regarding which words can appear after each other.

A speech recogniser trained to recognise flight captain's cabin speech might have a grammar network like shown in Figure 3.5. This kind of system could recognise easily utterances like

> *Good evening ladies and gentlemen, this is your captain for this flight Johanna Johannasdottir. We are flying at 8 kilometres over Reykjavik. We're expected to arrive at our destination on time.*

and

> *Good morning passengers. I am your captain Johan Johansson. We're flying at 24000 feet above the Arctic Sea. We're expected to arrive at our destination around 15 minutes late.*

A finite-state grammar recogniser has a list of internal states, and in each state it will wait for a certain type of input. An ASR telephone banking system might first be in a state, where it asks for the customer to give some kind of information. This might be given in various forms, including name or customer identification number, uttered in a sentence like "hello, this is ... calling" or "here is customer number ...". After establishing the identity, the recogniser might be in state, where it waits for the customer to declare the banking operation, like "What is my account's balance" or "I'd like to transfer $600 to the account ...".

At first, it might look like the number of possible sentences the speech recogniser has to be able to recognise is very large. And actually it is. But it is very small compared to free-form speech. Consider, for example, if the captain of the aeroplane took the bad microphone and started reading theatre reviews from old 1920's newspapers - how many of the passengers would understand it completely?

3.5.5 Large vocabulary continuous speech recognition

The most advanced task is **large vocabulary continuous speech recognition** (LVCSR). These kinds of recognisers have a vocabulary ranging from a few thousand to a few hundred thousand words. The recogniser tries to recognise whole sentences within a huge lattice of possible word combinations. The best of these systems, like the one in Figure 3.6, are capable of transcribing very complex utterances from a range of speakers.

Figure 3.6: A very complex speech recogniser in the dark future of Mega-City one. This device can recognise very complex sentences from different speakers. From: 2000 AD, issue 9, published by IPC Magazines.

This requires a statistical model for the language in order to be successful. This model is typically an **n-gram model**. The smallest n-gram model, the unigram, tells the probability of each word; a 2-gram tells the probabilities of two words appearing one after another. Larger n-grams tell the probability of longer word sequences.

In the decoding example above, even a small language model would tell us that the word sequence "I need" is more probable than "eye need".

The size of the model plays a very important part in the evaluation of the speech recogniser. Again, the size can be measured in several ways.

The most important factor is the vocabulary size. The more words, the more prone to confusion the model is. Another measure, the order of the language model is the n of the highest n-gram it contains. This is the length (in words) of the longest word sequence included in the model. A higher n means longer context and thus normally better recognition of sequences of words that the model already knows. Too much context can also be bad - If optimisation is not done carefully, the model might be bad for recognising more exotic sentences.

Yet another size measure is gram count - The number of different n-grams included in the model. This is controlled by pruning algorithms

A huge model with as many words as the reader can imagine poses a problem of memory consumption and recognition results with minor errors are more likely. Smaller models have a higher probability of making the right recognition, but at the cost of getting very bad results when a spoken word is not included in the vocabulary of the model.

Large vocabularies and inflectional languages

In English language, a selection of a few thousand words should be enough to cover one specialised subject area and a few hundred thousand enough to cover most of everyday speech.

However, in languages like Finnish or Estonian, the words are inflected according to their grammtic role, and all nouns and verbs can have up to thousands of different forms. As listing all the forms of the recognisable words is not computationally feasible, another technique is used to process morphologically rich languages.

Creutz, 2006 suggested splitting words into smaller units called morphs automatically in a way that optimises the number and the information value of the morphs. In this way long and uncommon words are split more than short and common words. What is common and what is not depends of course on the material used to train the morphing algorhithm. A morphing algorhithm trained on a collection of newspaper text would make the following division of the sentence *"Lopullisessa muodossaan sen hyväksyy kirkolliskokous"*:

lopu | *llisessa_* | *muodo* | *ssaan_* | *sen* | *hyväksy* | *y_* | *kirkollis* | *kokous_*

As shown by Hirsimäki, 2009, using morphs improves the recognition of inflectional languages, as long as the order of the language models is high enough to cover the increased number of units.

3.6 Acoustic modelling

We will now have a quick look at the last, but maybe the most fundamental part of speech recognisers, the acoustic models. The acoustic models try to describe the sounds of speech in the most (statistically speaking) accurate way.

Previously, we considered speech recognition as pattern recognition, and as a classification task. The acoustic models are the "boxes" into which the observations are sorted. Each phone has its own acoustic model, and this model tries to cover as many different ways of pronouncing this phone as the task requires, but not more.

Figure 3.7: Voice uniqueness is not a big deal for people, but computers easily detect the differences, like in the dark future of Mega-City One. From: 2000 AD, issue 11, published by IPC Magazines.

As with language models, where a larger vocabulary and more complex sentences lead to weaker results than a vocabulary and grammar tailored for a specific task, so it is with acoustic models. A more general model covers more different speakers, but the more general the model, the larger the possibility of confusing phones with neighbouring ones.

Each person's voice is unique, and while this is not such a big deal for people, for computers it can make a huge difference (like in figure 3.7). The question of model generalisation is of utmost importance. Speech recognisers that are optimised to recognise speech from one speaker, do it very well for this speaker and badly for everyone else.

This is the last part of the introduction to speech recognition, and unfortunately it seems to be inevitable, that I have to introduce decision tree clustering and mixture modelling in the following sections.

3.6.1 Hidden Markov Models

Hidden Markov Models (HMMs) bring a dimension of temporal flexibility to a recogniser. Markov model is a statistical framework for modelling probabilistic phenomena, where the **internal state** of the phenomenon changes irregularly according to some **transition probabilities**. This transition is checked at regular intervals after a predefined **time-step**.

The HMMs used in speech recognition have typically around 3-5 internal states,

that describe the acoustic signal of various parts of the phone, typically the beginning, middle and end part. The time-step is typically 5-15 ms - in this work it is 10ms, the same as the time-step of feature generation as mentioned in Section 3.4.

The transition table tells the probability of skipping to the next state of the phone at each time-step. The bottom image in Figure 2.4 shows a the sub-phone state segmentation using HMMs. Every 10ms the system investigates the new part of the acoustic signal, and knowing the state of the last 10ms, decides if the system should a) stay in the same state of the same phone, b) change to a new state of the same phone or c) change to the starting state of a new phone, if the transitions of the previous phone allow it.

The interested readers should satisfy their thirst for further knowledge on HMMs with Rabiner, 1989.

3.6.2 Phonetic context

If we recognise phones with no interest in their pronunciation context, we are recognising **monophones**. This would be all we need, if human articulatory system wasn't so dynamic. So far, we have observed in 3.6.1 that the acoustic signal of a phone is different at different times in the phone's lifetime. A second aspect of change arises from the relative slowness of the tongue and the lips – It takes a little bit of time to change the position of articulatory organs from the position required by one phone to a position required by another one. Using the knowledge – or at least educated guesses – about the previous and following phones, we can prepare for these changes and increase the accuracy of the recogniser. A model that takes into account one previous ('left') and one following ('right') phone is called a **triphone**. A model that looks at the two previous and two following phones is called a **quinphone**.

Taking the example sentence from 3.5.2, and running it through a large-vocabulary continuous speech recogniser, which uses simple acoustic models with no context, we get

se on varas kaspace dureta.

Finally, using a large language model and simple acoustic models that take into account the left and right phonetic contexts, we get

se aika raskas roseduuri että.

When using acoustic context, we have have two possible problems. First, we might have very little or no training data for some context-dependent phones that would

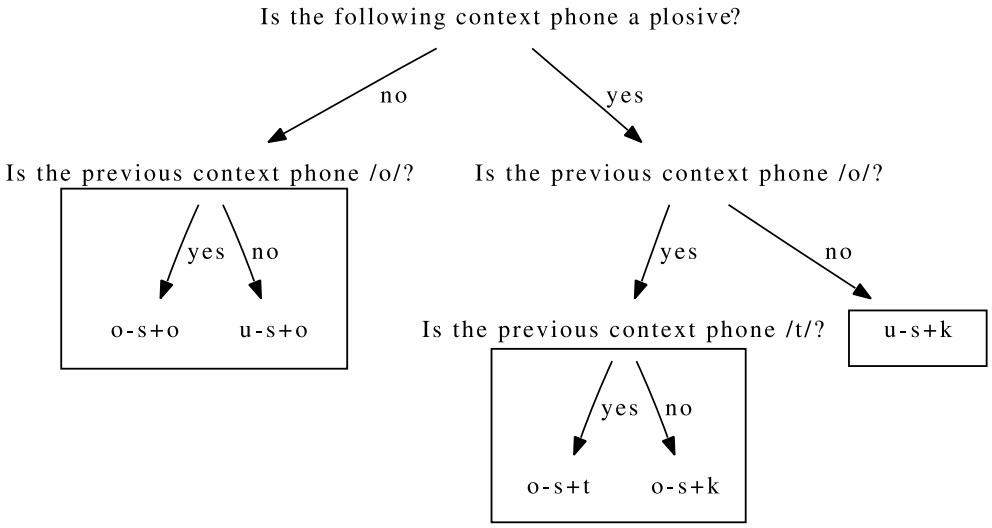


Figure 3.8: An example of a phonetic decision tree.

appear in the recognisable data. Secondly, we might end up with a very complex recogniser with so many parameters, that the recognition becomes slow.

3.6.3 Phonetic Decision Trees

The first problem can be broken down to two subproblems, and both are solved in a similar manner. With the help of a list of questions about the phonetic nature of the previous or following phones, we can form groups of similar context-dependent phones. For example, we want to be able to recognise the following triphones:

- o-s+k - not enough training data
- o-s+t - not enough training data
- o-s+o - no training data
- u-s+k - enough training data
- u-s+o - enough training data

Our list of questions has the following questions:

- Is the previous context phone a plosive?
- Is the following context phone a plosive?
- Is the following context phone /o/?

- Is the previous context phone /o/?
- Is the following context phone /t/?
- Is the previous context phone /t/?

With these we can build a decision tree, as shown in figure 3.8.

This list of questions is prefabricated by human hand according to previous phonetic knowledge. A full list, as shown in Table 3.1, has many questions dealing with known phonetic classifications. Our simple example pales in comparison, but hopefully yields some insight into the matter.

The tree is built to split with maximum information gain. Basically, each question on the tree should split the data into two subgroups, that should be as equal in size as possible. This is naturally done by the computer.

At the bottom of the tree, when no more questions are left to ask, are the phones themselves.

By looking at the tree, we can then see, which phones are supposedly phonetically similar. Then by traversing the tree from phones that are missing in the training data (like o-s+o) or have very little data (like o-s+k and o-s+t) we can use the tree to find suitable "partners" for these phones. So, the recogniser will use the model for u-s+o to recognise o-s+o. Also o-s+t and o-s+k are grouped. Together they might have enough training data, and share one model between them. If there still is not enough data, they would be grouped with u-s+k, and the three would use one model, trained with the data of all of the three phones.

3.6.4 Mixture modelling

Up until now we have avoided the question of the nature of acoustic models. A Gaussian Mixture Model (GMM)-based recogniser models the phones as sets of Gaussians. The most widely known Gaussian is the bell-curve function of the normal distribution. Each phone HMM's state has its own set of Gaussians to model the acoustic qualities of the spoken sound.

A gentle introduction to statistical modelling

A model in the sense of statistics is a simplified representation of data. It allows us to describe vast quantities of data with a few parameters. Of course, individualness is lost when describing the general properties of a larger population.

A very simple example of everyday use of statistical modelling is the use of average values. For example, to say that 1,5 million children younger than 5 years old die

Continuent	m_{en}, n_{en}, ng_{en}, f_{en}, v_{en}, th_{en}, dh_{en}, s_{en}, z_{en}, sh_{en}, zh_{en}, hh_{en}, l_{en}, r_{en}, y_{en}, w_{en}, m_{fi}, n_{fi}, f_{fi}, v_{fi}, s_{fi}, h_{fi}, l_{fi}, r_{fi}, j_{fi}
IVowel	ih_{en}, iy_{en}, i_{fi}
C-Central	t_{en}, d_{en}, n_{en}, s_{en}, z_{en}, zh_{en}, th_{en}, dh_{en}, l_{en}, r_{en}, t_{fi}, d_{fi}, n_{fi}, s_{fi}
Back-Stop	k_{en}, g_{en}, k_{fi}, g_{fi}
RoundFront	y_{fi}, oe_{fi}
Dental	t_{en}, d_{en}, n_{en}, t_{fi}, d_{fi}, n_{fi}
Front	p_{en}, b_{en}, m_{en}, f_{en}, v_{en}, w_{en}, iy_{en}, ih_{en}, ey_{en}, eh_{en}, p_{fi}, b_{fi}, m_{fi}, v_{fi}, t_{fi}, e_{fi}, i_{fi}, y_{fi}, ae_{fi}, oe_{fi}
Front-Fricative	f_{en}, v_{en}, f_{fi}, v_{fi}
EVowel	ey_{en}, eh_{en}, e_{fi}, oe_{fi}
Voiced-cons	jh_{en}, b_{en}, d_{en}, dh_{en}, g_{en}, y_{en}, l_{en}, m_{en}, n_{en}, ng_{en}, r_{en}, v_{en}, w_{en}, z_{en}, b_{fi}, d_{fi}, g_{fi}, n_{fi}, m_{fi}
Voiced-Stop	b_{en}, d_{en}, g_{en}, b_{fi}, d_{fi}, g_{fi}
NonCoronal	p_{en}, b_{en}, m_{en}, k_{en}, g_{en}, ng_{en}, f_{en}, v_{en}, hh_{en}, y_{en}, w_{en}, p_{fi}, b_{fi}, m_{fi}, k_{fi}, g_{fi}, f_{fi}, v_{fi}, h_{fi}, j_{fi}
NonStrident	f_{en}, v_{en}, th_{en}, dh_{en}, hh_{en}, f_{fi}, v_{fi}, h_{fi}
OU	o_{fi}, u_{fi}
KeskiEtu	ey_{en}, eh_{en}, oe_{fi}, e_{fi}
Unvoiced-cons	p_{en}, t_{en}, k_{en}, s_{en}, sh_{en}, f_{en}, th_{en}, hh_{en}, ch_{en}, p_{fi}, t_{fi}, k_{fi}, s_{fi}, h_{fi}
Palatal	k_{en}, g_{en}, ng_{en}, k_{fi}, g_{fi}
Long	iy_{en}, ow_{en}, aw_{en}, ao_{en}, uw_{en}
Syllabic	er_{en}, r_{fi}
Medium	ey_{en}, er_{en}, ah_{en}, ow_{en}, eh_{en}, e_{fi}, oe_{fi}, o_{fi}
UnStrident	p_{en}, b_{en}, m_{en}, t_{en}, d_{en}, n_{en}, k_{en}, g_{en}, ng_{en}, l_{en}, r_{en}, y_{en}, w_{en}, p_{fi}, b_{fi}, m_{fi}, t_{fi}, d_{fi}, n_{fi}, k_{fi}, g_{fi}, l_{fi}, r_{fi}, j_{fi}

Table 3.1: Examples phonetic questions used in the system. The questions are asked for every phone twice, for preceding and following phones. The group of phones, for which the answer is yes, are given on the right side. The phones used in the systems are described in the table 7.3.

of diarrhoea every year[1] is a statistical interpretation of a very large number of single cases happening over larger time span. It is considered true even if on some years only 1,4 billion die and on some years more than 1,6 billion die. So, our one-parameter model describes the phenomenon in enough detail to comprehend it, although not quite accurately enough to trigger desperately needed political action to stop it.

In speech recognition the models require more parameters in order to be effective. The speech sample data are represented in feature vectors, a list of parameters as described in the section 3.4. These vectors are simple ordered groups of numbers, often some dozens of numbers long. In the same way, a tailor's customer could be described according to a list of parameters: height (in cm), weight (in kg), waistline length (in inches), foot size (Continental standard) , etc. The result might be like [180.0, 75.3, 32.4, 43, ...].

Sometimes we can find more or less well defined groups in this kind of representation - maybe we could find correlations between some properties, and make crude predictions that we could sometimes make based on our everyday experience, like: If a person has a beard length of more than 0, the person is probably male.

Gaussian distributions and simple classification Given a large enough group of people, properties like height and weight are normally distributed - In large populations their distributions approach the normal distribution, which is a Gaussian distribution with the bell-shaped curve familiar to all of us.

The Gaussian distribution is defined by two variables that are easy to calculate from samples: mean and variance. The bell-shaped curve of the normal distribution is the **probability density function** (PDF). It shows how the probability mass is distributed across the observation space. Given that there are enough samples and that the phenomenon we are describing truly is normally distributed, the PDF shows the proportion of samples at a certain value interval related to the total mass of samples. Simple visual inspection shows that most samples are near the mean point of the distribution.

When we have several classes of samples, as an example 13-year old boys and 13-year old girls, and the distribution of their weights in kg[2], we can make a crude whether a new 13-year old of unknown sex is a boy or a girl, depending on the measured weight.

[1]From: Black Robert E, Morris Saul S, & Bryce Jennifer. 2003. Where and why are 10 million children dying every year? The Lancet, 361(9376), 2226 - 2234.

[2]From: Cynthia L. Ogden, Cheryl D. Fryar, Margaret D. Carroll, Katherine M. Flegal: Mean body weight, height, and body mass index, United States 1960 - 2002.

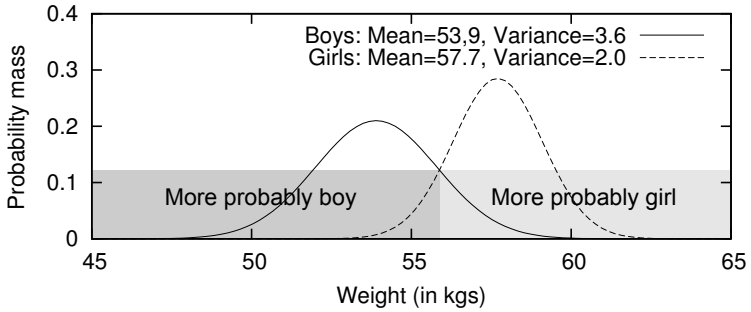


Figure 3.9: Weight distributions of 13-year olds. By drawing the probability distribution functions, we can see the point where they meet. Below this point, an unknown observation is more likely to be a boy, and above it, a girl.

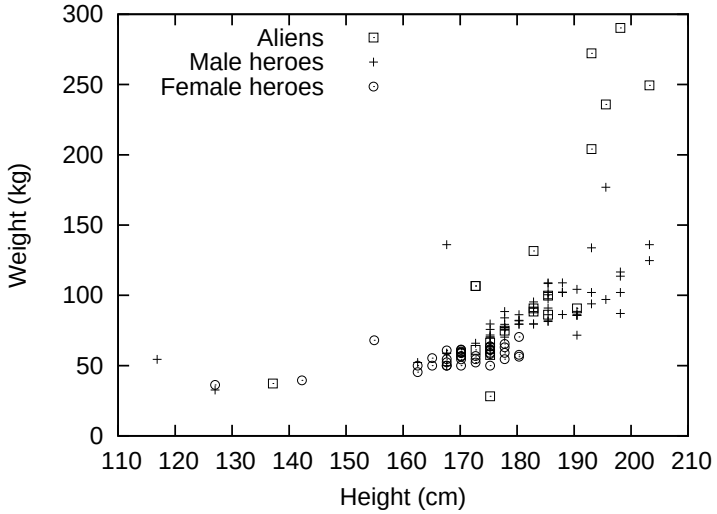


Figure 3.10: Height and weight of some Marvel super-heroes.

We can plot the PDF of the weights, and by comparing the values at the observed point, we can say which is the more probable class for the observation. This is illustrated in Figure 3.9.

Multi-variate Gaussian distributions Now, as an example, we will plot the heights and weights of 129 Marvel Comics' super-heroes and villains into one graph - including males, females and aliens[3]. This is shown in Figure 3.10.

We are interested in the powers of statistical modelling in classification problems. In order to classify observations, we start by modelling the previous observations.

[3]This is inspired by http://www.karenhealey.com/papers/comparative-sex-specific-body-mass-index-in-the-marvel-universe-and-the-real-world/. Data from the above and Marvel.com wiki.

We will fit one distribution to each of the three classes (Aliens, females and males), assuming that their properties are normally distributed. We will now try out different approaches to the models.

In the first case, we assume, that height and weight are **independent** variables, a distribution like this is described by its height and weight averages and their variances.

With the distributions fitted on to all the data classes, we can compare the probability density functions of the distributions. As mentioned earlier, the PDF describes how the probability mass is distributed across the observation space. The most probable classification is the one that gives the best probability for this observation point in its distribution i.e. When we compare the classes at a certain weight and height point, the most probable class is the one that has the greatest value in its PDF at that point. In Figure 3.11, all three PDFs have been drawn, and at the bottom are the original observations and the lines along which classification is done.

Looking at the classification boundaries in Figure 3.11, we can see that this model predicts that any super-hero weighing more than 80 kg and less than 120 kg is male. Heavier super-heroes are aliens, and the only interesting classification boundaries are at the 40-80 kg range, where height also plays a part in the prediction of super-hero type.

The model did not take into account the possibility of height and weight being dependent on each other - Often tall people weight more than short people. Maybe we should allow for this dependence to show. Thus, we will include in the model parameters the **covariance** of the variables. Having covariance between height and weight, we get the PDFs and classification borders as shown in Figure 3.12.

In this case both height and weight are important in guessing the super-hero type. A 140 cm super-hero weighing 150 kg would be alien, whereas a 210 cm 150 kg super-hero would be male.

This clearly gives a more accurate description of the data. However, another problem arises in practical applications: In a feature space with a large amount of dimensions, heavy calculation is required and a lot of parameters need to be estimated in order to build a robust model with **full covariance**. If there is no covariance, the term often used is “diagonal covariance”, as the covariance is usually shown in matrix form where the variances are on the diagonal.

To ease the calculation but still maintaining the ability to describe the data adequately, we will create each distribution from several Gaussians, which we will call components. The system compromising of several Gaussians is a **Gaussian mixture model**, balanced so that they describe the phenomenon as well as possible

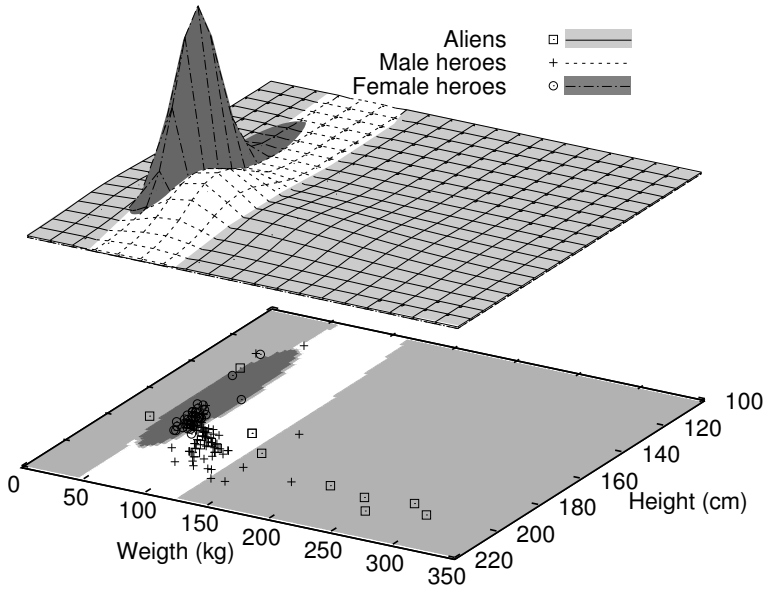


Figure 3.11: Example of multivariate diagonal covariance Gaussian probability distribution function. The distribution funtions have been drawn on top of each other, so that the color shown on the upper part of the figure is that of the most probable distribution.

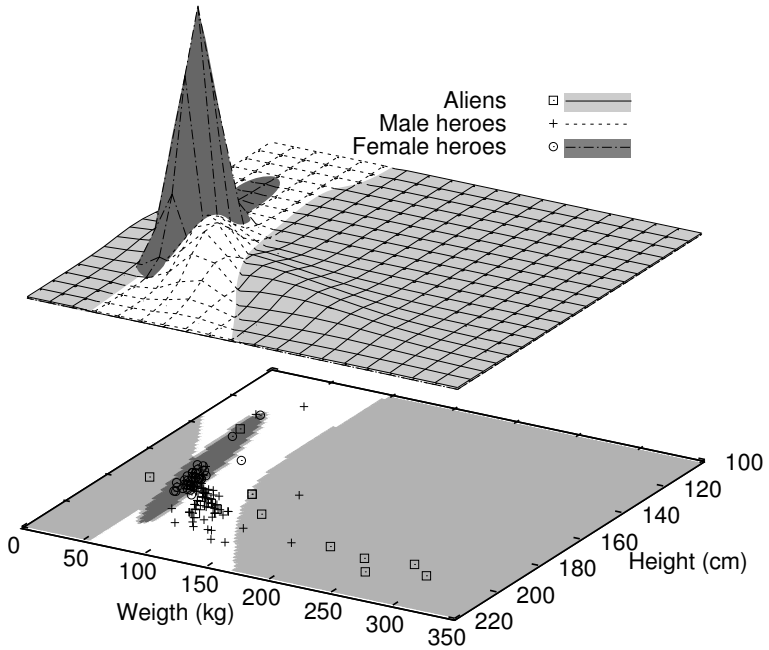


Figure 3.12: Example of full covariance multivariate Gaussian probability distribution function.

while their combined probability mass stays the same. These balancing factors are called **mixture weights**.

So now we have inside all the super-hero classes separate model for short and light super-heroes and a separate for tall and heavy - Or whatever the groupings are that our learning algorithm will pick out. The upper part of Figure 3.13 shows all the newly generated mixture models. All the models consist of 2 Gaussians. This models very well the existing super-heroes, and is very efficient in predicting the classification of similar data.

However, the female class is dangerously compact compared to the other classes. We could spread the distribution of female heroes a bit to make the model generalise better. This we can do by setting a **variance floor**, which indicates the minimum acceptable variance. Any variance component that is estimated below this value would be replaced with the minimum floor value. However, this could also be a sign that all Marvel's female super-heroes are very similar in build and thus flooring the variance would lead to unoptimal model.

By expanding the mixture model further by adding 1 Gaussian to the female model and 2 Gaussians to the male model, we get a pdf illustrated at the bottom of Figure 3.13. Now we have very complex classification boundaries, and this is maybe more an example of **over-fitting** the model on the data. Overfitting occurs when there is not enough training data to train all the model parameters properly. Normally, we would like to avoid too complex and too specialised models like this.

Using a-priori information If we know something before investigating the observation, this knowledge is "a priori". In this case we know that the observed super-hero comes from the same population as the heroes used to train the models. So, to further increase the accuracy of the classification, we can investigate the proportions of the classes in training material. Looking at the numbers, we calculate that 15% of the heroes were alien, 54% male and 31 % female. If we had to make a wild guess about the nature of the new super-hero, the best bet would be on it being male.

By balancing the PDFs of the three classes by these proportions, we'll end up with a slightly differing division of classes that places more bias on males and less on aliens.

This simple example tried to illustrate that with statistical knowledge, we can create '"boxes" for classifying new observations. Given the weight and height of a new super-hero, we can make a prediction whether the super-hero is male, female or alien. Similarly, in our 39-dimensional observation space for speech, a new observed

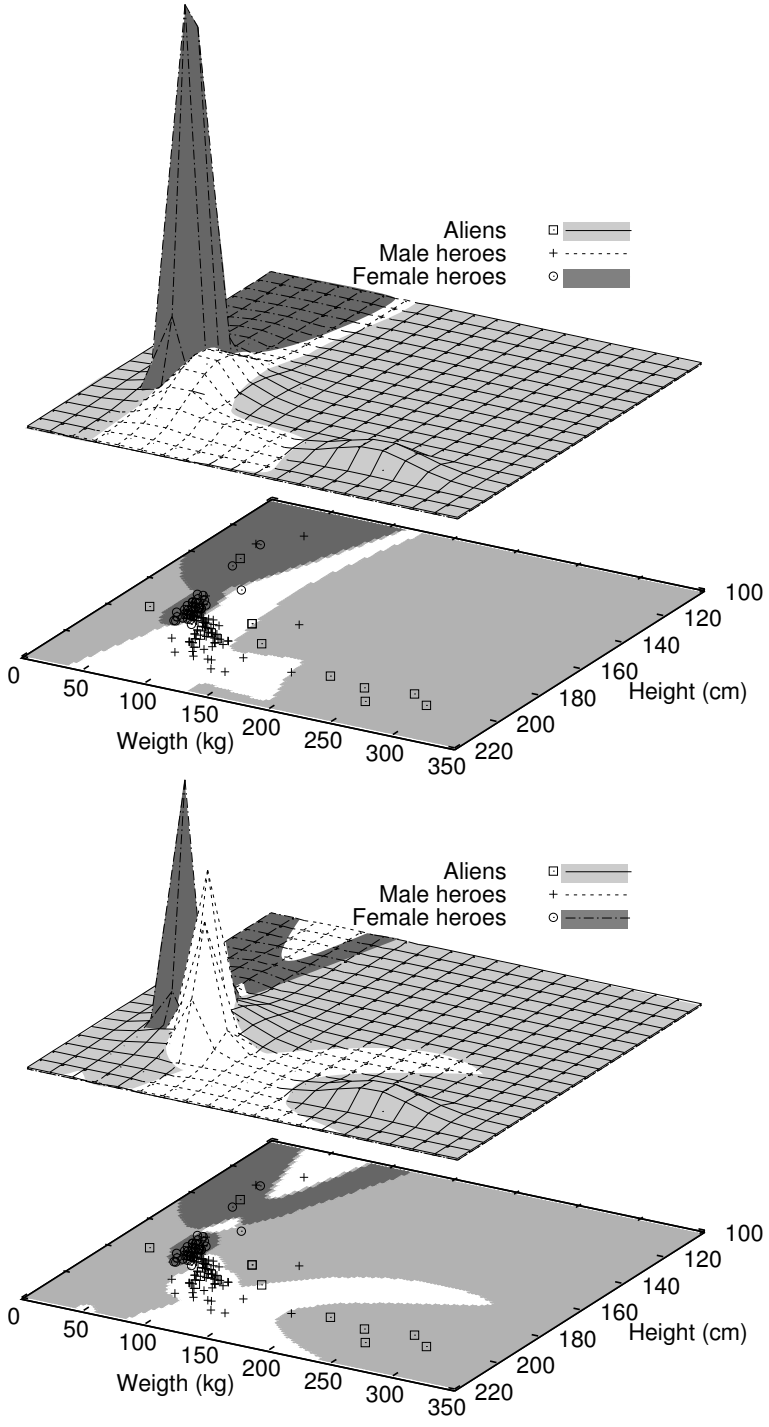


Figure 3.13: Example of probability distribution functions of a Gaussian mixture model with 2 Gaussians per class (above) and an overfitted model with 2,3 and 4 Gaussians per class (below).

and preprocessed sound datum can be categorised into one of the several thousands of classes that we have taught from large corpus of spoken data. Instead of adjusting the classification by simple a-priori information, we use language models or finite-state grammars to improve the classification.

The acoustic models in this work are diagonal covariance Gaussian Mixtures. The mixture approach is a simple and efficient way of increasing the performance of acoustic models. The above example from 3.6.2, recognised with a Mixture model of 2 Gaussians per state gives:

se on aika raskas roseduuri että

Semi-tied covariance transforms

One of the objectives of feature extraction is to create feature vectors where each item is independent of all others. This is very hard to accomplish, and thus there remains a problem of covariance modelling.

One possibility to model the covariance with a light computational load is to rotate the feature space with a transformation matrix. This rotation can be either class-specific or a general one. The classes can be quite arbitrarily defined - they can be individual states, phones or phone groups.

One approach to this is to create class-specific transforms for covariance matrices, **Semi-tied Covariance Transforms** (STCs). Thus, every model will have its own diagonal covariance matrix, and beside that, a class-specific transform is calculated so that the mismatch between transformed diagonal covariance model and full covariance model is decreased. For a detailed explanation, see Gales, 1999.

For the purpose of training STCs, a full covariance model set needs to be trained, and this is memory and processing time intensive but very rewarding when viewing recognition performance. The STCs used in this work are centre-phone-specific - for each centre-phone, there is one STC.

Chapter 4

Recogniser performance measurement

In this chapter we will quickly go through properties that can be used to compare the performance of different speech recognition systems.

As with any other technical tool, the "goodness" of a speech recognition system is evaluated by its performance in a given task. This can be compensated by considering the resources consumed by the system – an adequately performing cheap solution might sometimes be rated better than a system that performs very well but is too expensive, be it in Euros, Rupees or computation time, to ever be widely and effectively used.

The performance is generally measured with a predefined test set of sentences, words or sounds to be recognised. A scoring program is used to compare the recognition results (sometimes referred to as **hypothesis**) with the **reference transcriptions**. The reference transcriptions (sometimes referred to as reference **labels**) are usually generated by humans who write down what the test utterances contain, or if they are computer-generated, they have been checked by humans.

4.1 Recognition accuracy

Sometimes when we review several speech recognition systems, none of them can recognise the test utterances perfectly. For this case we need some kind of metric to compare which one does the job best. We'll review some of the more often used measurement methods, starting with accuracy.

Put simply, recognition accuracy tells us what proportion of recognition tests were successful.

$$Accuracy = \frac{H}{N} * 100$$

where H is the number of correct units in the hypothesis and N is the total number of units in the reference labels. This is a perfectly good way of measuring the performance of recognition of single words or phones - **word accuracy** or **phone accuracy**, and is normally not used for recognition of sentences or other continuous speech.

4.2 Word, letter and other error rates

For continuous speech, a better measurement method is to calculate the recognition errors rather than the aforementioned count of correct words.

The usual error comparison metric is based on the **Levenshtein distance** (Levenshtein, 1966). This is based on the minimum amount of editing operations necessary to change the correcr reference string into the erroneous recognition string. The operations are insertion, deletion and substitution. Once we have done the more complex calculation of the required operations, the calculation of the error is straightforward enough using the formula

$$Error rate = \frac{S + D + I}{N_r}$$

where S is the number of substitutions, D is the number of deletions, and I is the number of insertions that have to be made in order to reach the hypothesis from the reference. N is the number of units in the reference transcriptions.

We can calculate the error with different units depending on the language. Word error rate (WER) is a good measure for the English language, and is the *de facto* standard for comparing speech recognisers of most European languages.

WER is a simple and effective metric for languages where compound words are not very common and the mapping from words to phoneme sequences is not straightforward. In all cases it is not always ideal. For example, the recognition of an unknown word *"Phosphorescent"* as *"flows for a stand"* gives 1 substitution error and 3 insertion errors:

```
id: (rhpsi_ena06)
Scores: (#C #S #D #I) 7 1 0 3
REF:  so this is what it sounds like ***** *** * PHOSPHORESCENT
HYP:  so this is what it sounds like FLOWS FOR A STAND
```

```
Eval:                                    I     I   I S
```

From a human point of view, maybe this should be regarded as a single error, as it is only the absence of one word in the recognition vocabulary that has caused the erroneous transcription. However, word matching is a simple operation for a computer, whereas evaluating the similarity of word strings such as "phosphorescent" and "flows for a stand" is not.

Beside the clusters of errors caused by unknown words, a big problem is that mumbling and stuttering, sighing and laughing might be interpreted as words. The example output of a recogniser struggling with an utterance with some stuttering gives an idea of what might happen at worst:

```
id: (rhape_ena07)
Scores: (#C #S #D #I) 3 7 0 9
REF:  i READ    YOUR *** ** ** WEBSITE that **** ******* ******
HYP:  i HAVEN'T SEEN ONE A. M. AND     that THEY HAVEN'T SIGNED
Eval:   S       S    I   I  I  S            I    I       I


...  *** ********* YOU  ARE TWENTY SEVEN so **
...  AND SATELLITE THAT THE SYSTEM AND   so ON
...  I   I         S    S   S      S        I
```

For the 10 words of the original label, there is a total of 16 errors and thus an error rate of 160%. What exactly does an error rate above 100% mean? The recognition is not even 100% wrong in the eyes of a human reviewer - 3 words are recognised correctly. As an error rate this tells us only that for every word in the original sentence, 1.6 errors are made in the recognition process.

For short utterances and compound words, the case is even more extreme. Take for example the Finnish pronunciation for "1600", "tuhatkuusisataa". If the recogniser interprets this utterance as two separate words we have:

```
REF:   tuhatkuusisataa **********
HYP:   tuhat           kuusisataa
Eval:  S               I
```

From 1 substitution error, 1 insertion error and 1 word in reference, we get a word error rate (WER) of:

$$\frac{1+1}{1} * 100\% = 200\%$$

This seems a bit unfair - the recogniser made a small error that many people also make. Alternatively, we can break the sentence into single letters, using single characters instead of words as the base for calculating the errors, and then we'll have:

```
REF:   t u h a t * k u u s i s a t a a
HYP:   t u h a t _ k u u s i s a t a a
Eval:            I
```

From 1 deletion and 15 letters in reference we get a letter error rate (LER) of:

$$\frac{1}{15} * 100\% = 6,6\%$$

This seems to be a fairer measure for the error rate. LER is however not applicable to all languages. It requires a straightforward mapping between phones and letters, such as in the Finnish language.

For Asiatic languages, Character Error Rate (CER) is used instead of LER. For any language, it is also possible to use Phone Error Rate, as long as we keep track of recognised phonemes.

4.2.1 A few words about alternate hypotheses

It is not easy to define the "goodness" of even the perfect speech recognition system. Given a system that gives a 100% correct mapping of any speaker's spoken utterances to text in minimal time, the user might be slightly disappointed seeing the screen of the computer filled with the normal mumbling and stuttering associated with informal speech. To give an example, a hand-made transcription of the sentence used in the paragraph above, with as much of the acoustic information transcribed as possible:

umm I I read your web web website that you are twenty seven or so.

A more useful transcription would be a proofread version of the above:

I read your website that you are twenty seven.

However, as the original sentence clearly has human-audible extra words, we could allow the computer to hear these also. Various scoring software allow alternate transcriptions to be marked for each sentence. An example of this is the **trn -format** [1] where possible correct word sequences are marked inside brackets, separated with slashes:

[1] http://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/infmts.htm#trn_fmt_name_0

{ah / umm / @ } i { i / @} read your { web / @ } { web / @ } website that you are twenty seven { or / @} { so / @ }

The scoring program can expand this rather confusing looking sequence to a group of all possible sentences, and finds the best match for the original sentence. The @-character represents the empty word. If it is present inside the curly brackets, that means that this alternate can be left out completely if it fits the recognition result better. For example, both

ummm I read your website that you are twenty seven

and

I I read your web website that you are twenty seven so

are considered correct when using this method of alternative transcriptions.

4.2.2 Statistical significance

As speech recognition is an artful blend of signal processing and statistical mathematics, we are always concerned with unideal components in the signal and its processing, both when training and evaluating speech recognition systems. We call these unideal components **noise**, whether they are results of data recording problems or the sum of rounding errors and approximations in the computational operations. Therefore we must, when comparing two results that are very close to each other, ask if the difference is *statistically significant* or just a result of random fluctuation.

Matched pair test

The criterion for statistical significance depends on the test set size and type. Gillick & Cox, 1989, recommend to use a matched-pairs test to see whether the two recognition results are statistically significantly different.

The test runs as follows: First, the evaluation utterances are recognised by both the recognition systems A and B. For each utterance i in the n utterances in the evaluation set, N_{Ai} is the error score by the recognition system A, and N_{Bi} is the error score made by the recognition system B. The error score is WER, calculated as described in section 4.2. The test variable is the difference of the error counts in each utterance i, defined as $Z_i = N_{Ai} - N_{Bi}, i = 1, 2, ..., n$. An estimate of the variance of Z is

$$\hat{\sigma}_z^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Z_i - \hat{\mu}_z)^2 \tag{4.1}$$

where $\hat{\mu}_z = \sum_{i=1}^{n} Z_i/n$. If we define

$$W = \frac{\hat{\mu}_z}{\hat{\sigma}_z/\sqrt{n}}, \tag{4.2}$$

and have $n > 50$, W can be considered as normally distributed, and we can determine statistically significant differences in two different recognition results of the same sentences by a standard P-test on W.

4.3 Language Identification performance

A multilingual recogniser can be used for language identification, especially when using combined language models as in Weng *et al.* , 1997. This could be one tool for evaluating a system, that needs to know the language spoken by the user in order to generate output in the corresponding language. In this thesis, this is not considered relevant.

4.4 Performance in a task

A speech recognition system deployed for consumer use is evaluated simply by its success rate in the task it is given. These metrics are very task specific. Some examples could be:

- An ASR telephone meteorologist would be judged by its ability to give the correct weather forecast to the dialling customer.
- A telephone banking ASR, that has a zero margin for error and double-checks every input from the user, could be rated by the time it requires to make the correct banking transaction.

4.4.1 Keyword recognition

A speech recogniser can be used to index audio data of radio and TV programs, lectures, speeches etc. in order to do quick information retrieval. Instead of recognising all sentences correctly in the data, it is more important to find the relevant keywords that the users might be interested in. In this case, evaluation methodology of information retrieval systems should be used for evaluating the system.

Figure 4.1: A speech recogniser user interface in the dark future of Mega-City One. Its usability depends very highly on its ability to pick keywords from the user's speech. From: 2000 AD, issue 9, published by IPC Magazines.

This kind of performance metric is necessary for various user interface tasks. A voice-controlled kitchen lighting system or the system in Figure 4.1, should only respond to specific keywords, not to small-talk or general chatter.

4.5 Other factors

Sometimes the raw result numbers from tests are not quite enough to guide us when deciding which recogniser is the best for our use. The results depend heavily on the quality of test data, the computational power of the test system and the overall complexity of the test tasks.

4.5.1 Real-time Factor

All of the phases of the recognition require some kind of optimisation between speed and accuracy. Brute force is unfortunately not always a real possibility. Scientific research systems might not be time-critical, but recognisers aimed as user-interface tools for consumer use need to respond to input very quickly.

The real-time factor is a ratio between the computation time required to process an utterance and the length of the utterance.

This naturally varies according to the computing system used to run the recogniser. If a recognition system has a real-time factor of 1.0 when running on supercomputer, but is to be deployed on a small mobile device, some performance problems are to be expected.

The RT factors in this work range from 5 to 20, depending on the complexity of the used language models.

4.5.2 Acoustic scalability

All people have slightly differing voices. Some people have very unique ways of speaking. Children sound different from adults, men different from women. This is also an important factor in noise robustness. The microphone might be bad or far away. The background noise can vary by time.

Also there is a question of how accurately we want to recognise the nuances of the voice. We could have more parameters in the acoustic models to recognise lexical stress, but the more models we have, the greater the risk of confusing one sound with another. The fine analysis of different sounds is rarely necessary, computationally heavy and must be accompanied by a very highly refined language model. This would make the whole system slow and might lead to more errors.

4.5.3 Vocabulary size and utterance complexity

Depending on the purpose of the speech recognition system, the vocabulary and complexity of the recognition tasks can vary enormously. Recognising a few simple utterances like

> *lights on*

and

> *lights off*

from a list of a dozen commands is computationally much simpler than recognising dictated sentences that are grammatically complex and consist of comparatively rare words, like the English sentence:

> *New York City's Fresh Kills landfill on Staten Island for one dumps four million gallons of toxic liquid into nearby freshwater streams every day* [2]

or the Finnish sentence:

> *jiddisiä puhuva rabbi ei ollut aiemmin maistanut jaffaa.* [3]

[2] From the WSJ0 corpus
[3] From the Speecon corpus.

In the case of the language model, the most important choice the builder of a speech recogniser has to make is the size of the vocabulary. The more words, the better the result for rare words, but the worse the recognition rate of the most common words.

Obviously, these two extreme examples require different properties from the speech recognition systems. The language model component especially needs to be fit for the task. An idea of the aptness of the language model in the task of recognising a particular set of sentences is given by the perplexity of the model calculated over the reference text.

The perplexity is a metric for describing how well the set of labels fits the language model. It is based on how well a language model can predict the words in recognisable sentences. Perplexity is calculated by

$$PP = \hat{P}(w_1, w_2, ..w_m)^{-\frac{1}{m}} \tag{4.3}$$

where $\hat{P}(w_1, w_2, ..w_m)$ is the probability of the word sequence $w_1, w_2, ..w_m$ (a sentence with a length of m) according to the language model.

Perplexity helps to explain why our results are better or worse than would be expected considering the quality of our acoustic models.

A more critical and even more obvious consideration is the question whether all the words to be recognised are included in the recogniser's dictionary. The number of out-of-vocabulary (OOV) words gives a lower limit for the recognition error. Consider the example sentences in Section 4.2. As our recogniser does not have the word "Phosphorecent" in its vocabulary, the recogniser is bound to make at least one error. This is only the lower bound, and in our example the recogniser makes 4 errors while struggling with the word.

4.5.4 Utilitarian value

No engineering science is completely cut off from the world, and therefore it is good to contemplate on the effects of speech recognition innovations on general human happiness.

In the footsteps of Mill, 1861, we might ask how much this or this speech recognition system will have positive impact on lives of humans.

First, considering the experiences of consumers, we might claim that any increase of performance in speech technology leads to an improvement of the lives of the people already using speech technology.

Often a speech recognition system might be added to a telephone service system

for two reasons: It saves costs and increases capacity – Less wages, shorter queues.

So advances in speech technology lead to call-centre wage-slaves becoming redundant and thus having to start job-hunting again. This might be a real problem for the older generations, just a few years from their deserved pensions. On the other hand, it might be seen that technology frees these people from the shackles of a tedious job. Thus the effects of speech technology in the lives of the ex-employees of a telephone service depend on the possibility of attaining a better way of life after an honourable discharge from service at the company.

For the user, who will confront a computer rather than a human in the other end of the telephone line, the experience might improve or degrade. The computer will never tire and will never make human errors. A bad system can be hopeless to use and will decrease the happiness of the user.

Impacts like these should be pondered upon when deploying systems that will replace employees.

4.6 Conclusions

Speaker-independent large vocabulary continuous speech recognition is the Champions' League of speech recognition. With a large vocabulary, free-form grammar and considerable acoustic variation in test speakers, the error rates will be higher than with systems with a reduced number of users and strictly defined test tasks.

For this thesis, I will build different speaker-independent recognisers and compare them to existing ones. For English, measuring the word error gives the best comparison independent of the task. For Finnish, I will use letter error.

Chapter 5

Multilingual Speech recognition

In this chapter, we will familiarise ourselves with the basics of multilingual speech recognition and go through some selected results in this field.

5.1 Basic concepts

The goal of multilingual speech recognition research is to create speech recognition systems that can be used to recognise speech from several languages without retraining or changing the system in between when switching languages.

A multilingual system can either be a combination of several language-specific systems or a single system capable of recognising all the input languages. For the acoustic models this means either selecting the proper model set at recognition time or building model sets that can recognise two or more languages.

In this chapter and some of the following, I will give examples of relationships of phonemes of one language to phonemes of another language. Underscore abbreviations are used to indicate the language to which a phone belongs. Thus a marking /ah$_{\mathrm{A}}$/ describes the phoneme /ah/ as being present in language A.

5.2 Recogniser porting

Before delving into the question of multilingual recognition, a practical trick to train a new speech recogniser in a new language has to be mentioned.

In order to create an initial acoustic model set for a new language, we can use the acoustic models of a recogniser meant for another language to create preliminary phonetic labelling for training data, and therefore evade flat-starting the models from scratch.

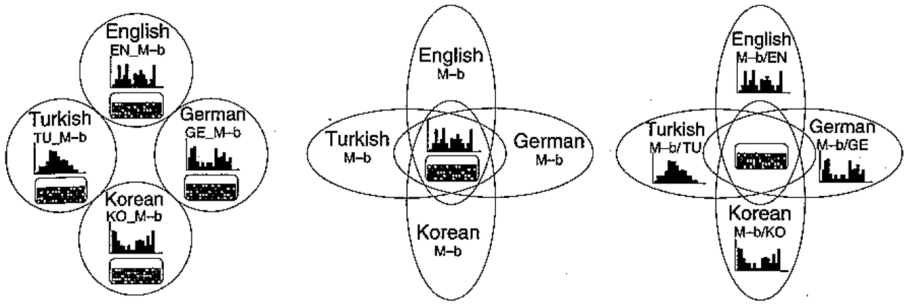


Figure 5.1: Different approaches to combining data across languages in acoustic models. From left to right: ML-Sep, ML-mix and ML-tag. From Schultz & Kirchhoff, 2006.

Sometimes, for languages that are "similar enough", a cost-effective way to build an acoustic model set for a new language is to use an existing model set of a recogniser of a similar language as a basis. As building a robust speaker-independent recogniser requires hundreds of hours of accurately transcribed speech, whereas porting an existing recogniser to a new language requires only some dozens of hours, the decision should be easy enough from an economic perspective.

Multilingual recognition can then be done using the new and old recognition systems as one combined system in the ML-sep way, as will be explained in the next section.

5.3 Parameter sharing

To enable true multilingualism in a single recogniser, the acoustic models of the involved languages have to be somehow merged or trained together. The depth of sharing acoustic models can vary from including two or more sets of language-dependent acoustic models in the same model set to sharing most of the phone models between phones of two languages. As defined by Schultz & Kirchhoff, 2006, we can identify three different approaches to parameter sharing: ML-sep, ML-tag and ML-mix, as shown in 5.1.

5.3.1 ML-sep: Separate models for different languages

When we need to use acoustic models from several languages in a single recogniser, the simplest approach is to load all the trained models into the recogniser as separate entities but to handle them as one.

What these model sets have in common is the preprocessing process of the data. If dimensionality reducing transforms (LDA, HLDA) are used, all the models must be considered in its calculation.

5.3.2 ML-mix: Phone sharing

In ML-mix approach, the phones of different languages that are assumed to represent the same "universal" phone, are pooled together for training, and for all computational considerations, their lingual origin is forgotten. In context-clustering, it is considered to be a member of all the languages where it appears.

Supposing, for example, that the ASR system builders considered both /a_{fi}/ and /ah_{en}/ as a representation of the IPA phone *a*. In a ML-mix system, there would be a single /a/-model shared by both languages, When recognising the Finnish word "Akuutti" with its pronunciation

Akuutti /a_{fi}/ /k_{fi}/ /u_{fi}/ /u_{fi}/ /t_{fi}/ /t_{fi}/ /i_{fi}/

or English word "acute" with its pronunciation

Acute /ah_{en}/ /k_{en}/ /y_{en}/ /uw_{en}/ /t_{en}/ .

both words would have the same /a/ in their pronunciation dictionary form:

Akuutti /a/ /k_{fi}/ /u_{fi}/ /u_{fi}/ /t_{fi}/ /t_{fi}/ /i_{fi}/

Acute /a/ /k_{en}/ /y_{en}/ /uw_{en}/ /t_{en}/.

Up to which extent the phones are shared has to be carefully thought. A careless unification of sound units across languages degrades performance, as evident in the test results presented in Chapter 8.

5.3.3 ML-tag: Gaussian sharing

One interest in sharing acoustic models in a Gaussian mixture system is to reduce the number of Gaussians in the system, and thus reduce computational complexity at runtime.

The number of Gaussians per state has a profound effect on the recognition result as well as computation time. When the models of two languages are brought together, the natural question to ask is, are some of the Gaussians similar enough to be shared among the systems?

The ML-tag parameter sharing scheme uses the same Gaussian components across languages for the same phone. However, the mixture weights are not shared across languages. Instead, the weights are trained separately for each phone in each language.

5.3.4 Rule-based and data-driven combinations

The mentioned approaches require some sort of rule for combining phones across languages. An abundance of combination approaches are available, but basically all are based on some kind of clustering in the feature-space, sometimes complemented by phonetic rules. The phonetic rules are often based on the IPA classifications of phones of each language.

5.3.5 Phone combination metrics

Clustering of phones requires a distance measure. When this calculation is done in feature-space, the Gaussian representations of the emitting HMM states can be used as a basis. A number of distance metrics are available, most notably Bhattacharyya and Kullback-Leibler distance, which both take into account variance of the Gaussians. For a comparison of distance measures, see Sooful & Botha, 2002.

Kullback-Leibler divergence

The similarity metric in this work is always based on the Kullback-Leibler (KL) divergence. This divergence, as proposed in Kullback & Leibler, 1951 tells how well one distribution of data corresponds to another. The divergence is calculated by computing the integral of a supposed probability distribution function (pdf) of observations over the pdf of a reference distribution:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} P(x) log \frac{P(x)}{Q(x)} dx \tag{5.1}$$

where Q is the reference model and P is a model of observations.

In this work there is no reference and observation; both distributions are hypotheses, and as KL divergence is non-symmetric, to get a proper distance metric, we will simply calculate the distance in both directions and add together:

$$Distance = D_{KL}(P||Q) + D_{KL}(Q||P) \tag{5.2}$$

In the case of multivariate Gaussian distributions G_p and G_q of d dimensions, the

KL divergence has an analytical form of:

$$D_{KL}(G_p||G_q) = \frac{1}{2}(log_e\frac{|\Sigma_q|}{|\Sigma_p|} + Tr(\Sigma_q^{-1}\Sigma_p) + (\mu_q - \mu_p)^T\Sigma_q^{-1}(\mu_q - \mu_p) - d) \quad (5.3)$$

For a multivariate Gaussian mixture model, there is no analytical solution to the function. We can estimate the divergence with Monte Carlo method of random sampling. According to Hershey & Olsen, 2007, with n samples we have:

$$D_{MC}(G_p||G_q) = \frac{1}{n}\sum_{i=1}^{n} log\frac{G_p(x_i)}{G_q(x_i)} \rightarrow D_{KL}(G_p||G_q) \quad (5.4)$$

as $n \rightarrow \infty$. This estimation becomes rather cumbersome for context-dependent Gaussian mixture models with numerous Gaussians.

5.4 A review of existing systems

Multi-lingual experiments have been done in various scales from single phone recognition to large vocabulary continuous speech.

The first proposal for combination of acoustic models of several languages by Andersen *et al.* , 1993, was a data-driven combination of monophone models of Danish, British English, German and Italian. The similarity measure was based on phone confusion. The language-independent phoneme recogniser used in their tests got a 37.6% recognition accuracy, a slight decline to the language-dependent 36.4% baseline.

An intermediate approach is to share the same pool of Gaussians for the different languages. This was investigated between English and Swedish in Weng *et al.* , 1997. Of the varied array of tests, some yielded better results than the baselines. One reason for this might be the small amount of training data per language.

Gokcen & Gokcen, 1997, created acoustic models from the phones of US English, French, and Japanese and tested it on sets of 10-15 words and short phrases by more than 1000 speakers from each of the following languages: US English, Brazilian Portuguese, French, German, Japanese, and Spanish. Overall recognition rates were more than 92% for each language. English recognition improved with the addition of foreign training data (91% - 92%), Japanese got slightly worse (98% - 96%).

Imperl & Horvat, 1999, combined Slovenian, German and Spanish acoustic models by a data-driven approach based on confusion, with triphone confusions calculated by combining the confusion of the three monophones of each triphone. In best case

System	Lngs.	Merge style [1]	Parameter reduction [2]	Test type	Performance change
Andersen *et al.* , 1993	5	Conf.	Mono -33%	Word RR	-3.2%
Gokcen & Gokcen, 1997	3	Clus.	Mono -70%	Word RR	*
Imperl & Horvat, 1999	3	Clus.	Global -64.7%	Word RR	-3%
Imperl *et al.* , 2003	3	Clus.	Tri -40%	Sent./WER	-1.67%
Vihola, 2001	5	Clus.	Mono -70%	Word RR	-6.7%
Kumar *et al.* , 2005	3	Mixed	Mono -54%	Word RR	-0.6...-3.4%
Schultz & Waibel, 1998	5	Mixed	Global -60%	Sent./WER	-1.2... - 5%
Lyu & Lyu, 2008	2	Clus.	*	Sent./SRR	+7.8%

[1] Confusion based, Clustering based, Phonetic rules based or Mixed
[2] Reduction in percentage of number of monophones, number of triphones or all parameters

Table 5.1: Summary of existing multilingual ASR systems. The table lists the number of languages, HMM geometry type, parameter merging technique and reduction, test type and change in performance of best presented system relative to baseline systems.

they managed to compress the size of acoustic models by 64.7% with less than 3% decrease of word accuracy rate and 1% decrease of language identification rate.

Later in Imperl *et al.* , 2003, by using agglomerative clustering, a reduction of more than 40% in the size of the acoustic models could be reached with a recognition degradation of only 1.7%.

A combined English, Finnish, German, Italian and Spanish acoustic model was investigated in Vihola, 2001. The baseline language-dependent recognition systems had a WRR of 89.0%. With no phone clustering, a system with 105 SAMPA phone models had an average word recognition rate (WRR) of 84.6%. When this was reduced to 64 phones, the WRR dropped to 83.1% with Knowledge-based clustering or 82.2% - 84.4% for data-driven clustering methods.

Kumar *et al.* , 2005 combined the acoustic models of Tamil, Hindi and American English, with language-dependent baseline recognition rates of 98.5, 96.6 and 98.6 respectively. An end result of 95.1, 96.0 and 96.4 was reached with an initial hand-made phonetic division and a following data-driven clustering based on Mahalanobis distance. The set of 149 phones of the combined three languages was compressed to a set of 72 phones.

Lyu & Lyu, 2008, investigated combining a large Mandarin and a small Taiwanese corpus. They demonstrated that a 2-step clustering approach with agglomerative

hierarchical clustering and delta Bayesian information criteria is a useful data-driven method to generate data-driven rules to constrain the context-dependent phone clustering in an unbalanced bilingual corpus. From the best baseline systems employing model complexity selection, with syllable recognition rates (SRR) of 59,7% and 61,3% for language dependent and language independent respectively, they could reach a recognition rate of 64,4% by optimising the phone set.

In Schultz & Waibel, 1998, a LVCSR system for 5 languages was trained. Compared to the monolingual baseline systems, there was an increase of 20%-30% in word error rate in the multilingual systems. The phone sharing was done according to the IPA chart with a set of 82 phones for the multilingual systems.

A summary of the existing systems can be found in Table 5.1.

5.5 Literature conclusions

Based on the literature, the question " can we enhance the recogniser by combining acoustic models of several languages?" is relevant only when the languages are similar and at least one of the training corpora is so small that a monolingual recogniser would suffer from lack of data.

A drop in accuracy will be inevitable. Therefore a better question to ask is: "will the combined recogniser still be fit for the task?"

Chapter 6

Adaptive speaker-independent recognisers

In this chapter I will introduce the concept of adapting acoustic models of a speech recogniser and then present some techniques for adaptation. Then I will introduce cross-lingual adaptation.

6.1 Adaptation of acoustic models

The amount of training data is a crucial factor when it comes to creating world-class speech recognisers.

Training a **speaker-dependent** (SD) speech recogniser adequately requires some hours of accurately labelled training data. For obvious reason, this makes it very difficult to create a new speech recogniser for every customer in a speech-driven automated service – The reader can spend a moment thinking how it would feel to repeat sentences for a few hours into a phone number query service's automated telephone service before being allowed to ask for a number.

Instead of training a speech recogniser for every voice, we can build a generic, **speaker-independent** (SI) speech recogniser. We train it with at least a few dozen hours of spoken data from dozens or hundreds of speakers and average the results to get a recogniser for the most average of voices. This recogniser works well for people who do not stand out from the crowd when talking. For those with a funny or otherwise special voice, it works pretty badly.

The human brain is extremely adaptive to differences between speakers and acoustic environments. Children and women speak with a higher tone than men, people from different parts of the country speak differently. People speak faster, slower,

louder, more quiet, mumbling or clearly. Sometimes they stutter, sometimes shout. Our natural speech processing systems can cope with quite a bit of speaker variation, whereas the ASR system with a static model set has desperate time trying to recognise speech from a speaker with a peculiar way of speaking.

As training a speech recogniser is nothing more than fitting predefined model geometry to the acoustic features of the training speakers' speech, why not continue the same way, fit the models with their geometry into the recognisable speech?

We'll create separate adaptation **transforms** for every speaker or similar group of speakers, and apply these fitting transforms to the models or the features only when we recognise that persons speech.

The transform is a predefined mathematical operation that manipulates either the acoustic models or the features. Different transformations require a different amount of new parameters to be estimated, and thus increase the complexity of the recogniser system.

6.2 Speaker, noise and microphone adaptation

Even the speaker-dependent recognisers benefit from an adaptation mechanism. Adaptation can help against background noise or temporary vocal changes, caused by, for example, a flu or the previous night spent singing, shouting and drinking.

In an ideal setting, the training data and evaluation data come from a similar source where the microphone is identical and the recording conditions are similar. For speech recognition based services aimed for the general public over a cellular phone networks, the speech submitted to the recogniser might be recorded with any of the available terminal devices from different manufacturers, and depending on the network, might be encoded differently depending on the available bandwidth for the end user. The recording condition might be noisy; maybe the service is used while walking on a busy street or in a restaurant. The characteristics of the user's voice might thus change because of the transmission system and background noise. An adaptive system will then adjust itself to cover the distortion caused by the noise and microphone.

All in all, the adaptation not only helps in recognising utterances from new speakers, it also helps to recognise utterances from old, familiar speakers who record phrases in new conditions.

An interesting question is then, when we encounter a new speaker in new recording conditions, what part of the possible improvement by adaptation is attributed to speaker adaptation, what part to microphone adaptation and what part to recording

conditions adaptation?

In research cases, where the recognition evaluation datasets generally stay the same, this problem arises when the training and evaluation data sets originate from different corpora. This problem arises particularly in the test setup in the second part of this thesis, where recognisers trained with corpora A and/or B are used to recognise utterances from corpora C.

6.3 Supervised and unsupervised adaptation

Adaptation is similar to training the recogniser. When we know the transcription of a piece of acoustic data, we use the data to find a transformation that fits the speech data into the recognition models.

In **supervised adaptation** we have control over the adaptation procedure, so that only correct labels are used. In laboratory environment we have prelabelled data that we use to calculate the transformations for the models. Outside the laboratory, the same can be accomplished by asking the user to utter a few predefined sentences in order to "calibrate" the recogniser. Adaptation is then done on the assumption that the user really said what the application asked.

When it is not possible to use labelled data, **unsupervised adaptation** is used. Now the adaptation utterances are first recognised and then the acoustic data and the generated labels are used to calculate adaptation transforms. This process is illustrated in Figure 6.1 Obviously, if the recognition goes wrong, adaptation can also go wrong, but the threshold for this kind of misbehaviour is surprisingly high.

If a system continues to adapt itself, and the amount of data used to calculate the transformation accumulates, the adaptation usually slowly converges to an optimum. An example of this can be seen in figure 6.2.

Regarding usability issues, unsupervised adaptation offers a lot more possibilities than the supervised adaptation procedure.

6.4 Linear Transformations

A linear transformation is a simple matrix multiplication operation. Linear transforms can be computed from small amounts of data and can easily be updated as more data becomes available. A linear transformation is applied as a

$$\hat{\boldsymbol{x}} = \boldsymbol{W}\boldsymbol{x} \tag{6.1}$$

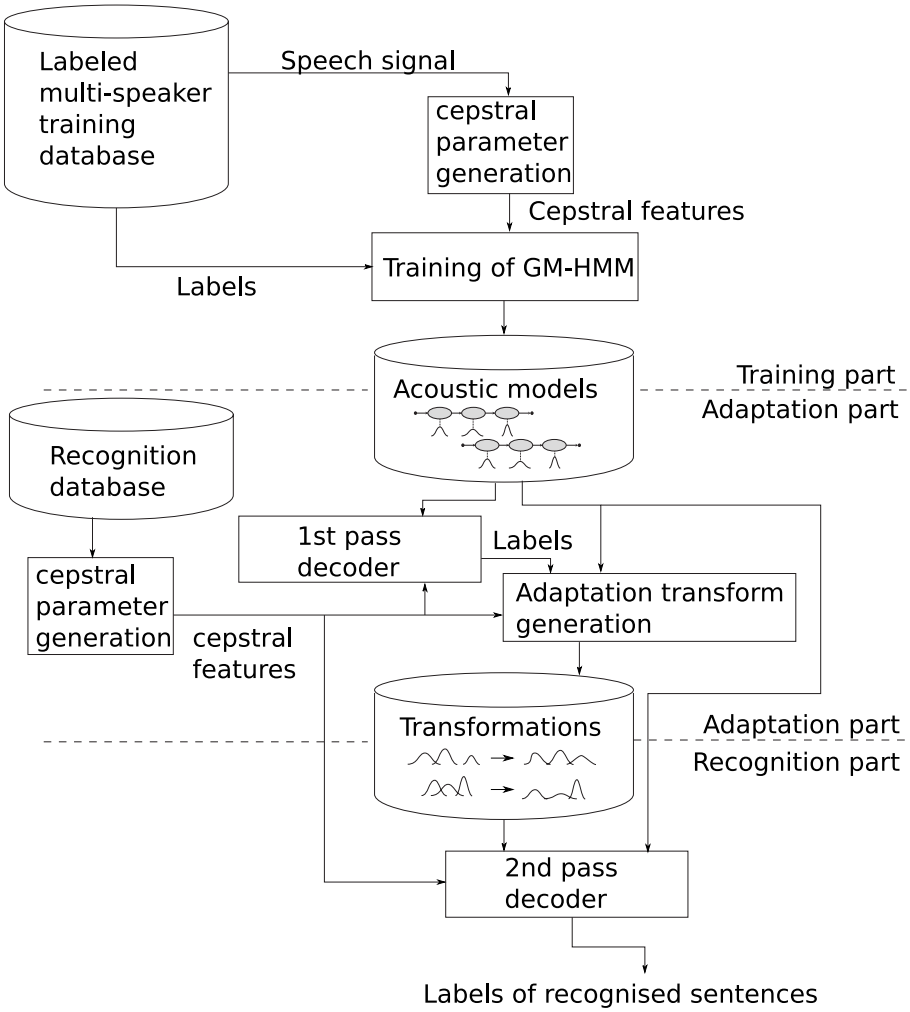


Figure 6.1: Two-pass recognition system using unsupervised adaptation.

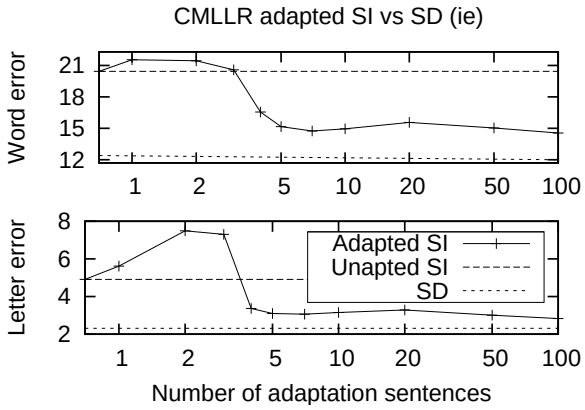


Figure 6.2: An example of the convergence of unsupervised adaptation. The first few sentences are recognised badly and cause the adapted system to perform worse than the unadapted speaker-independent system. This is more a special case, Figure 6.3 shows the average case.
SI system is trained from Speecon corpus, SD system from 2000 sentences of a female non-professional speaker. The utterances used for adaptation are different from the test utterances.

where the transformed variable $\hat{\boldsymbol{x}}$ is calculated by multiplying the original variable $\boldsymbol{x}$ by the adaptation matrix $\boldsymbol{W}$. This is the same for all linear transformations, whether transforming the mean vector $\boldsymbol{\mu}$, covariance matrix $\boldsymbol{\Sigma}$ or the feature vector $\boldsymbol{\zeta}$.

Given the trained acoustic models and some labelled training data, we can calculate a transformation for the models, so that the model will fit the training data as well as possible. These transforms that give the best fit to the data are called Maximum-Likelihood Linear Regression (MLLR) transforms. An MLLR transform of the mean values is often called MLLRmean, and the transform of covariances is MLLRcov.

For a linear transform of the mean values of the Gaussians of the acoustic models, we have the new mean values $\hat{\mu}$ given by a transform A:

$$\hat{\boldsymbol{\mu}} = \boldsymbol{A}\mu + \boldsymbol{b} = \boldsymbol{W}\xi \tag{6.2}$$

where ξ is the extended mean vector $[1\mu^T]^T$. The transformed Covariance matrix $\hat{\Sigma}$ is given by the a transform H:

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{H}\Sigma + \boldsymbol{H}^T \tag{6.3}$$

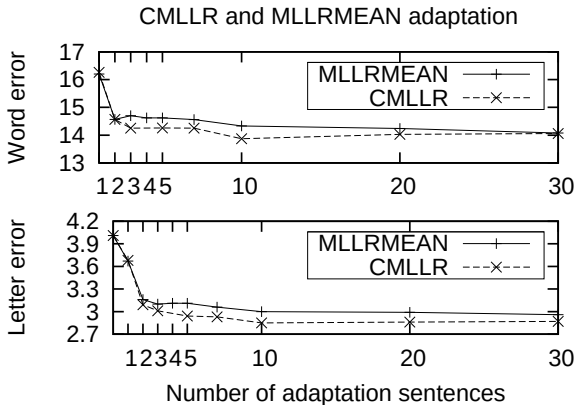


Figure 6.3: A comparison of cMLLR and MLLRmean adaptations, using a varying amount of sentences for adaptation. The adaptation sentences are included in the test set. The result is calculated from an average of 40 speakers.

In constrained MLLR (cMLLR) as presented by Gales, 1998 the transformation applied to the variance corresponds to the transform applied to the mean:

$$\hat{\mu} = \boldsymbol{A}\mu - \boldsymbol{b} \tag{6.4}$$

$$\hat{\Sigma} = \boldsymbol{A}\Sigma\boldsymbol{A}^T \tag{6.5}$$

Figure 6.3 shows a test of the two different linear transformations. Here cMLLR slightly outperforms conventional MLLRmean. The figure also shows how in the average case the quality of the transform rapidly improves over the first few sentences, and then improves very slowly.

As Gales, 1998 finds, full MLLR and cMLLR perform similarly. However, cMLLR adaptation is a simpler procedure than the cascade of MLLRmean and MLLRcov, and therefore in this work only cMLLR is used.

For those interested in the mathematical formulation of the linear transforms, the relevant formulas are wonderfully collected together in Young *et al.* , 2006.

6.5 Regression trees

Applying a simple transform on the whole feature space is simple enough. We can calculate a transformation matrix that can be applied to each phone model.

A major challenge is to apply different transformations to different phones. As phones are represented by various Gaussian distributions in different areas of the

cepstral feature space, we'd like to apply different transformations to different areas of the feature space. Ideally, we would like to create transformation parameters individually for each triphone for each speaker.

As this is not technically feasible – the data requirement is simply too high – we will create clusters of similar triphones that should be affected by the same transformation. Creating the adaptation transforms thus begins with creating the adaptation clusters, or rather, a regression tree of adaptation clusters (Gales, 1996). As we are constantly walking the thin line between over- and under-training, we will only increase the number of available adaptation clusters as we gather more data to be used for the adaptation. So, we start with one general adaptation matrix, and only use several matrices when we have collected a few sentences of adaptation data.

6.6 Normalisation and adaptation in feature extraction

Feature normalisation means preprocessing speech data so that it appears more uniform to the training and recognition systems. Some normalisation methods are very simple, like removal of DC coefficient from the output of a AD-converter.

Another normalisation method, **cepstral mean subtraction (CMS)** subtracts a long-time average of each cepstral coefficient channel from all values of that channel, reducing the effects of any static noise components like microphone distortion or steady background noise.

Some normalisation methods are done with different parameters to every speaker and thus can be considered adaptations. An example is **vocal tract length normalisation (VTLN)**, which warps the frequencies in filter bank analysis in order to compensate for different frequency distributions in the speech of different speakers.

Some of these noise and speaker compensations could be accomplished with MLLR-based adaptation, and so normalisation and adaptation can be seen as competing techniques. But even though MLLR adaptation becomes relatively less efficient when applied to models trained with more refined features (Pye & Woodland, 1997), it still improves the results and presently there is no reason not to use it whenever the computational resources allow.

6.7 Limits of adaptation

When using adaptation, depending on the corpus and tasks, some of the speakers are likely to show degradation in performance levels. Sometimes even 15% of the

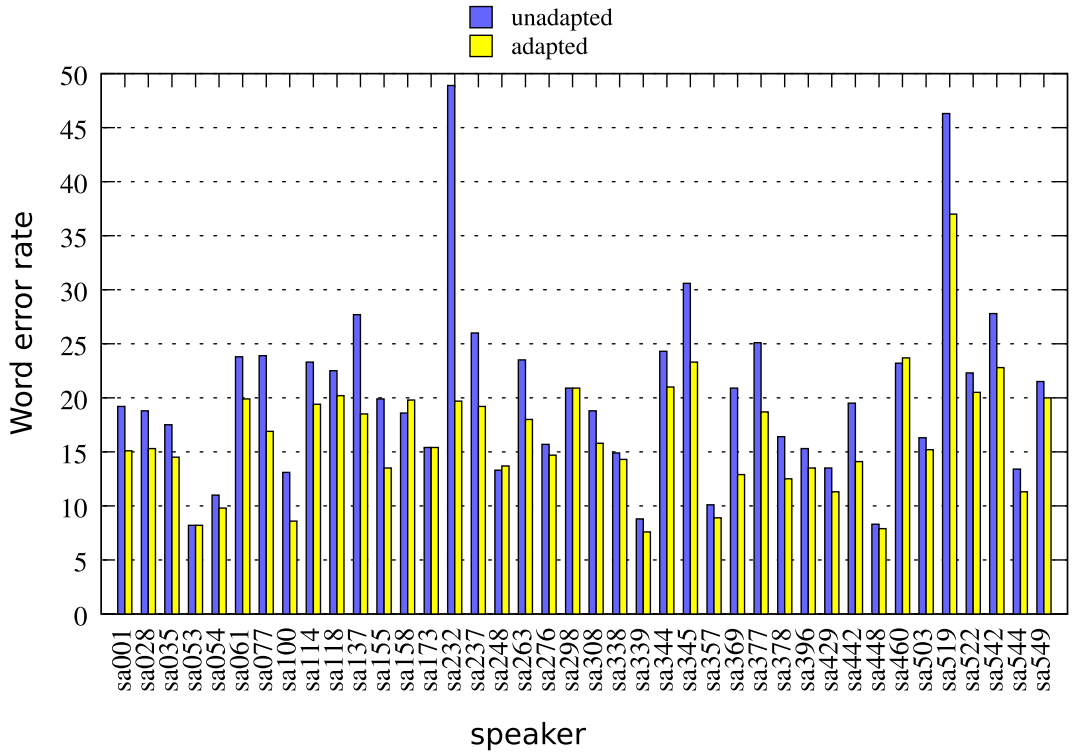


Figure 6.4: Speaker-specific word error in unadapted and adapted Finnish ASR. For each speaker, the left column shows the unadapted and the right column the adapted recognition results.

speakers show a drop in the performance when adding adaptation to the system (Mandal *et al.* , 2006). Figure 6.4 shows word error improvement of supervised CMLLR adaptations over unadapted system. Adaptation is applied to the 40 speakers of the Speecon test set. A vast majority of them show considerable improvement over the unadapted system, but a few - speakers *sa158*, *sa248* and *sa460* show slight degradation for the adapted system.

Also, adaptation is not something that will save the day when even when the recogniser is badly built. There are major restraints on the effectiveness of simple adaptation mechanisms. If the adaptation had no such restraints, we would, with enough data, reach the recognition accuracy of a speaker-dependent recogniser. This is not possible with the techniques of today. But we can get close enough to be able to build systems that are good enough for acceptable consumer quality restricted grammar tasks.

With the standard adaptation algorithms, we are only changing model parameters, not the geometrical frame of the model. So we have a predefined number of Gaussian mixtures in a predefined vector feature space, and with adaptation we can only change the form and location of the mixtures, not create new ones or change mixture weights. Also, we have predefined the triphone set with its phone clustering dependent on acoustic qualities or availability of data for the average speaker's voice.

It is possible that Maximum A Posteriori (MAP) adaptation could improve the simple adaptation schemes used in this thesis somewhat. Speech synthesis speaker adaptation techniques, like the one presented by Yamagishi *et al.* , 2009, use MAP adaptation to copy voices by adapting an average speech synthesis voice. MAP adaptation requires more data than MLLR adaptation, and is therefore not as attractive to application in ASR field.

6.8 Cross-lingual adaptation

In this thesis, cross-lingual adaptation means using the speaker's speech in one language to adapt the recogniser to the same speaker's speech in another language.

Cross-lingual adaptation forms an essential part in **speech-to-speech translation** devices, where the synthesised voice output (**Text-To-Speech**, TTS) of the device should resemble the voice of the original speaker. In this case, however, the adaptation is done on the synthesised output voice, based on the input to the recogniser, rather than the ASR component.

Cross-lingual adaptation for speech recognition is an experiment with seemingly

little practical value. The situations where the same individual would alternate between languages while using a speech recognition based user interface are relatively hard to imagine. Maybe a spying device that listens to your telephone while you are talking to agents of secret services of different nations, using different languages while having very short conversations. Short for the reason that a robust MLLR-based systems needs no more than 3 sentences to generate near-optimum adaptation transforms, as can be seen in Figure 6.3. This means that if so much data is available in the actual language to be recognised, the use of cross-lingual ASR model adaptation can hardly be motivated in consumer applications.

A far-fetched idea would be use several acoustic models to decode speech that has foreign names. Maybe the utterance *"I'd love to visit Versailles before heading to Charles-De-Gaulle Airport"* could be recognised with a mixture of English and French acoustic models (and a very exotic language model!) and in such a case the adaptation should be transferrable from one model set to another.

A parallel application that uses similar techniques is the cross-lingual speaker adaptation of speech synthesis.

Chapter 7

Test setup

After the introduction to multilingual speech recognition and various adaptation methods, I can now define the tests associated with this thesis with more precision. I will first define my hypothesis, then the testing arrangements and proceed to describe the recogniser training resources and methods. Then I'll continue about the practicalities of multi-lingual training and present a short analysis of the scope of multilinguality in the trained systems. Last, I will give recognition results for the baseline systems used in this work.

7.1 Hypothesis

I will train a set of recognisers, each with a single acoustic model that combines both Finnish and English language, and investigate their performance in normal recognition tasks and adaptation tasks, of which some are cross-lingual in the following way:

Data recognised in language L_A is used together with acoustic models of L_A to generate speaker-specific CMLLR adaptation transforms for speaker s in language A, T_{As}, and these are then used to improve recognition of speaker s's utterances in language L_B. In a shared acoustic model set $L_A = L_B$. This is illustrated in figure 7.1.

The hypothesis is, simply put, that the more of the acoustic models are merged, the more the overall recognition rate degrades, but at the same time, the more helpful adaptation data from one language is when used to help recognition of another language. As of now, there is no hypothesis which merging strategy would be the most fruitful: Whether it is the monophone combination or the triphone merging, remains to be seen.

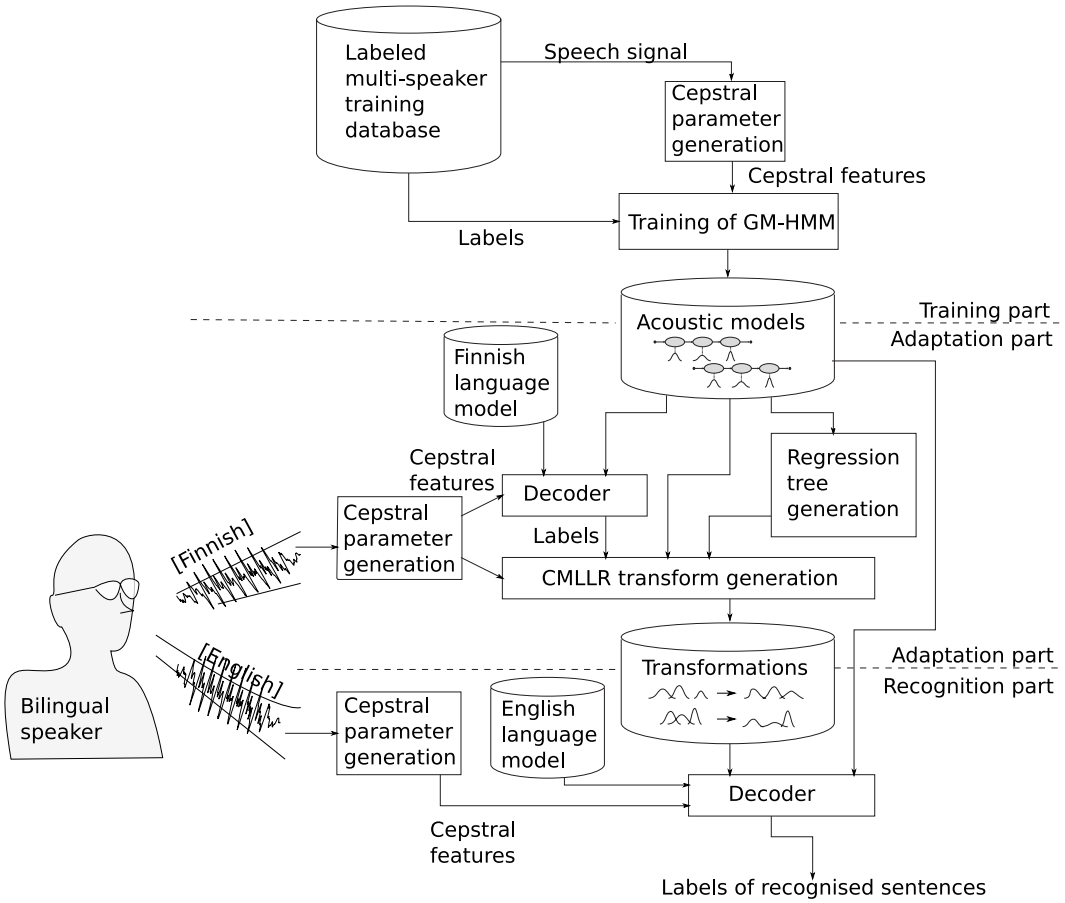


Figure 7.1: Principles of cross-lingual ASR adaptation.

7.2 Test preparation

Before running the tests, the test recognisers have to be trained.

All the recognisers systems used in this thesis are large vocabulary continuous speech recognisers based on Hidden Markov Models (HMM). They use the HTK standard mark-up languages for storing the acoustic models and ARPA format for language modelling.

They are speaker-independent systems and have one general model (average voice) each.

Each system has a pair of language models: A bigram for generating word lattices from acoustic data with acoustic models, and a higher-order model for expanding and rescoring these lattices.

Two-pass recognition is possible with CMLLR adaptations.

7.3 Resources

A selection of corpora were available for training and testing purposes. The Finnish Speecon and the American English Wall Street Journal 0 corpora were used for training.

The evaluation sets of the respective corpora were used for the evaluation of the systems. In addition, a small bilingual corpus was used to compare intra- and cross-lingual adaptation.

7.3.1 Training corpora

Speecon

The Speecon project, as described in Iskra *et al.* , 2002, is an international effort at creating collections of speech samples in a large variety of languages. The aim is to collect enough material to train speech recognisers for user interfaces. This means that the aim is to have a good enough representation of speakers to create true speaker-independent acoustic models.

The corpus includes material from both adult and child speakers. There is an emphasis on the phonetic diversity and structured utterances at the cost of neglecting spontaneous speech. The corpus is collected in several predefined recording environments, with an array of four microphones and various levels and types of background noise. The acoustic data is sampled at 16kHz, 16-bit mono PCM.

The Finnish Speecon database includes utterances from 550 speakers each uttering

around 30 complete, phonetically rich sentences plus a set of utterances including number sequences, answers to questions and spelling out words. Of these, utterances of 160 speakers are recorded in noisy environments and are left out of this thesis. 1118 sentences from 40 speakers are used for testing, 1093 sentences another 40 speakers are used to optimise training and recognition parameters, and 17178 utterances from the remaining 310 speakers are used to train the recogniser.

WSJ0

Wall Street Journal 0 corpus of American English as described in Paul & Baker, 1992, is used to train the English acoustic models. The corpus consists of numerous speakers reading aloud complete sentences from the Wall Street Journal newspaper. Similarly to Speecon, the recording conditions are carefully specified and the readers represent a wide enough range of voices to enable training of speaker-independent speech recognisers.

The default training set for speaker-independent acoustic models includes 83 speakers. With sentences that include out-of-vocabulary words removed, 7736 sentences are used for training. The default test set includes 213 sentences from 10 speakers. The acoustic data is sampled at 16kHz, 16-bit mono PCM.

The WSJ corpora are widely known and therefore among the most popular corpora to be used as test material for new developments in the field of speech recognition. Furthermore, the corpus distribution includes language models and a scoring software suite.

Depending on the model alignment, the training set of the Speecon corpus includes around 6 000 000 frames of spoken speech. The training set of the WSJ0 corpus has around 17 000 000 frames of speech.

The size differences between the corpora deserve a mention. With the phones sets used in this work (the CMU set[1] for English and the TKK phone set for Finnish) there are an average of 90 000 frames for each HMM state of each phone in Speecon and around 150 000 frames for each HMM state of each phone in the WSJ0 phone corpus. The amount of training data available for each phone can be seen in Figure 7.2. However, as there is a wider selection of phones in the English phone set, there is only an average 14000 training frames of each context-dependent HMM state in WSJ0, as opposed to an average 27500 frames for each state of each context-dependent HMM state in the Speecon corpus.

Because of this and as verified by experiments, the WSJ0 corpus requires a some-

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict#phones

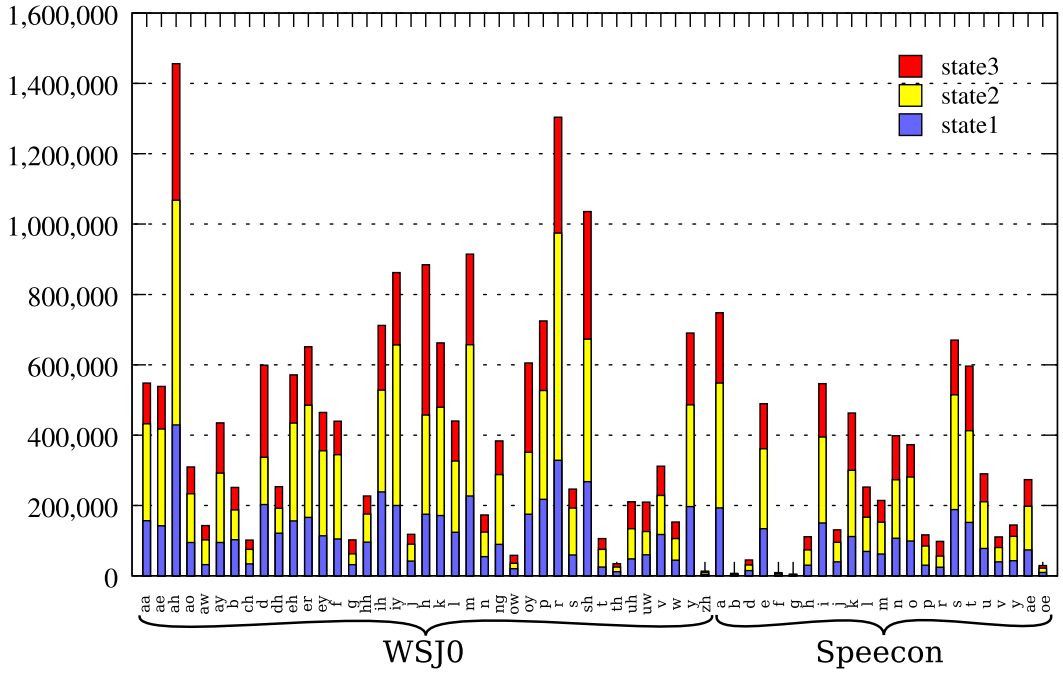


Figure 7.2: Frame and state distribution of frames in training corpora. Calculated from final monophone alignments of baseline systems.

what higher splitting threshold for building triphone tying decision tree. The higher threshold means less splits, and therefore more data per cluster and so more robust models.

7.3.2 Evaluation corpora and tasks

Some corpora distributions include predefined test sets to standardise recogniser evaluation. The WSJ corpus includes carefully defined test sets. These task sets focus on specific tasks for evaluating various aspects of LVCSR, like adaptation and speaker generalisation. The corpus also includes previously built ARPA language models, both bigrams and trigrams of various sizes (5000, 20 000 or 64 000 words, 2-grams and 3-grams) to be used for these tasks.

The main test set for this work is the speaker-independent evaluation test set with 5000 words vocabulary and no verbalised punctuation (nvp/si_et_05). This is an "easy" test set by any LVCSR testing standards, where similar WERs of 5-7% for unadapted systems have been reached already 15 years ago (Aubert *et al.* , 1994).

The Finnish Speecon corpus has been divided into training, development and evaluation sets. The division is the same as used at TKK where a Finnish recogniser was previously developed (Hirsimäki *et al.* , 2006).

The TKK recogniser has reached a LER of 3.3% with ML models (comparable to "Vanilla" models used in this work). With discriminative training methods, this has dropped to 3.0% in Pylkkönen, 2009.

Beside these two massive corpora, a collection of sentences recorded in both Finnish and English was used.

Table 7.2 summarises the properties of the development and test sets used in this work.

Bilingual EMIME sentences

A collection of 130 sentences in Finnish and 130 sentences in English were recorded by three Finnish speakers for tests on inter-lingual speaker adaptation in the EMIME project, with main emphasis being on synthesis. These sentences include:

- 100 phonetically rich Finnish sentences from Speecon corpus prompts
- 100 English news text sentences from the WSJ corpus prompts
- 25 sentences from the European parliament corpus in Finnish
- 25 sentences from the European parliament corpus in English

- 20 semantically unpredictable sentences in Finnish
- 20 semantically unpredictable sentences in English

The semantically unpredictable sentences are meant for the evaluation of speech synthesis, and serve no purpose in the evaluation of ASR systems of this work, so these are taken straight out when preparing the test set. Furthermore, the European parliament sentences do not fit the language models so well, and are dropped.

Also, sentences with Out-of-vocabulary (OOV) words are taken out of the test set. Thus, the test sets consist of 300 sentences in Finnish and 258 sentences in English, with 0 OOV.

The recordings were done in a silent environment with a high-quality microphone array. In this thesis these sentences are used for cross-lingual adaptation tests. The data is 16kHz, 16-bit PCM.

7.4 Language modelling

All the systems have 2 different language models. The recogniser setup first needs a bigram model to create lattices of most probable words. The decoder cannot use higher order language models, so this step requires a bigram model. A higher-order n-gram is then required to expand and rescore the lattices, and the final recognition result is derived by a Viterbi-search from these expanded lattices.

The WSJ language models are included in the corpus distribution. The Finnish language models are trained identically to the TKK recogniser (Hirsimäki, 2009). A slight exception is the replacement of word break morph with word breaks glued to the suffix morphs. This is required by the differing decoding software.

The recognition of the is done with the associated 5k-vocabulary language models using both the 2-gram and 3-gram language models.

The Finnish EMIME sentences are recognised with the same language model as the Speecon evaluation sentences. The English EMIME sentences are recognised with the WSJ 20k-vocabulary language models (2-gram and 3-gram).

Table 7.1 lists the properties of the language models. The language model fit to the test set data is described by the perplexity of the model on the given data set, as shown in table 7.2

7.5 Feature extraction

Feature extraction follows the outline presented in Section 3.4.

Language	Vocabulary	Gram order	Gram count
English	5000 words	2-gram	840 000
	5000 words	3-gram	4 300 000
	20000 words	2-gram	1 400 000
	20000 words	3-gram	6 700 000
Finnish	44000 morphs	2-gram	14 000 000
	44000 morphs	10-gram	23 000 000

Table 7.1: Size of language models.

Test / Evaluation sets

Task	Sent.	LM vocab.	Gram	OOV	Perplexity 2/n-gram
Fi Speecon test	1118	44k morphs	2/10	0	217 / 151
En WSJ0 si-et-05	330	5k words	2/3	0	110 / 57
Fi EMIME set	375	44k morphs	2/10	0	317 / 227
En EMIME set	258	20k words	2/3	0 / 0.11%	316 / 232

Development sets

Task	Sent.	LM vocab.	Gram	OOV	Perplexities
Fi Speecon devel	1054	44k morphs	2/10	0	210 / 146
En WSJ1 H2 P0	215	5k words	2/3	0.32%	106 / 62

Table 7.2: Perplexities and OOV rates of development and test sets. For the n-grams in last column, n=3 for English and n=10 for Finnish.

Corpus data is sampled at **16KHz**. Features are extracted from **25ms windows** at **10ms intervals**. This is more or less *de facto* standards for feature extraction in ASR (as used by Zigelboim & Shallom, 2006, Deshmukh *et al.* , 2002, Kinjo & Funaki, 2006, Hirsch & Pearce, 2000, ...), and optimising these parameters is outside the scope of this thesis. The sample edges are smoothed with a Hamming window.

The waveform is then transformed into a frequency representation by taking a Fourier-transformation of it. This frequency form is then processed with a Mel-frequency filter-bank, which leads to 22 coefficients.

A cosine transform is used to derive more decorrelated values, **cepstral coefficients**. As most of the information is packed into the first coefficients, I.e. lower

quefrency , only the first 12 of the 22 coefficients are used in the recogniser training and recognition processes.

Cepstral mean subtraction as explained in Section 6.6 is used as a normalisation method to decrease the interference of microphone differences and recording conditions between the various corpora.

The exact parameters used for feature extraction are listed in table 7.4.

7.6 Acoustic model geometry

The acoustic models are **3-emitting state, left-to-right HMMs** with Gaussian Mixture Model (GMM) emissions. These are explained on more detail in Section 3.6. The HMMs represent context dependent **cross-word triphones** i.e. the model depends on the previous and following phones, and context over word break is taken into account.

All model sets use **first and second order derivatives** of the features along the static features.

The silence models for all languages are context-independent long silence /sil/ (counts as a context, but does not take contexts into account itself) and context-free short silence /sp/ (not used as a context). Table 7.3 lists all the phones.

The careful reader will notice, that the Finnish phone set does not include diphthongs and does not make a difference between short and long sounds. In the Finnish system, long sounds are represented by contexts – in order to recognise a long /a/, we simply look for a sequence of two phones.

A standard way to describe the context dependent phones is to write the main phone in the middle, the preceding phone before it separated by a "-"-sign and the following phone after it, separated by a "+"-sign. I will use a "*"-sign to denote "any phone". Also, in triphone context I'm dropping the slashes for in order to increase readability.

So, in the case of a long Finnish /a/, we look for a "*-a+a" followed by a "a-a+*", as in the pronunciation of the Finnish word "Vaara":

Vaara *-v+a v-a+a a-a+r a-r+a r+a-*

7.7 Acoustic model adaptation methods

A 3-block CMLLR adaptation is used. The large 39 x 39 transformation matrix is divided into three 13 x 13 matrices ("blocks") that are on the diagonal matrix.

English			Finnish		
Phoneme	Example	Translation	Phoneme	Example	Translation
AA	odd	AA D	a	rotta	r o t t a
AE	at	AE T	ae	mäyrä	m ae y r ae
AH	hut	HH AH T	b	bertta	b e r t t a
AO	ought	AO T	d	daavid	d a a v i d
AW	cow	K AW	e	eemeli	e e m e l i
AY	hide	HH AY D	f	faarao	f a a r a o
B	be	B IY	g	gaselli	g a s e l l i
CH	cheese	CH IY Z	h	herne	h e r n e
D	dee	D IY	i	paavi	p a a v i
DH	thee	DH IY	j	jouhi	j o u h i
EH	Ed	EH D	k	kala	k a l a
ER	hurt	HH ER T	l	kala	k a l a
EY	ate	EY T	m	mäyrä	m ae y r ae
F	fee	F IY	n	noppa	n o p p a
G	green	G R IY N	o	rotta	r o t t a
HH	he	HH IY	oe	yö	y oe
IH	it	IH T	p	paavi	p a a v i
IY	eat	IY T	r	rotta	r o t t a
JH	gee	JH IY	s	sää	s ae ae
K	key	K IY	t	rotta	r o t t a
L	lee	L IY	u	ukko	u k k o
M	me	M IY	v	paavi	p a a v i
N	knee	N IY	y	mäyrä	m ae y r ae
NG	ping	P IH NG			
OW	oat	OW T			
OY	toy	T OY			
P	pee	P IY			
R	read	R IY D			
S	sea	S IY			
SH	she	SH IY			
T	tea	T IY			
TH	theta	TH EY T AH			
UH	hood	HH UH D			
UW	two	T UW			
V	vee	V IY			
W	we	W IY			
Y	yield	Y IY L D			
Z	zee	Z IY			
ZH	seizure	S IY ZH ER			

Table 7.3: The English and Finnish phones used in the monolingual recognition systems.

One block covers the original MFCC features, the second block the 1st order derivatives and the final block the 2nd order derivatives of the features. By using 3 distinct blocks instead of one large, we can reduce the computational time required by matrix multiplication.

Parameter	Value
TARGETRATE	100000.0
WINDOWSIZE	250000.0
USEHAMMING	T
PREEMCOEF	0.97
NUMCHANS	26
CEPLIFTER	22
NUMCEPS	12
ZMEANSOURCE	T
DELTAWINDOW	2
ACCWINDOW	2
USEPOWER	T
ENORMALISE	T

Parameter	Value
MINVARFLOOR	0.05
HADAPT:MAXXFORMITER	100
HADAPT:MAXSEMITIEDITER	20
Required occupation (triphone tying)	200
Tying threshold (triphone tying)	1000
Required training samples	10
Training beam	350.0
Number of Gaussians per mixture	16

Table 7.4: Fixed parameters for feature extraction (above) and acoustic model training (below).

7.8 Acoustic model training procedure

The training procedures of the baseline and test recognisers is kept as identical as possible. The initial training procedure follows that outlined in the HTK book (Young *et al.* , 2006), after which single Gaussian models are slowly grown to 16 Gaussian mixtures repeating the procedure of splitting mixtures and retraining, using 5 iterations of the Baum-Welch embedded training algorithm. The details of the training algorithms are outside the scope of this thesis, and the interested reader

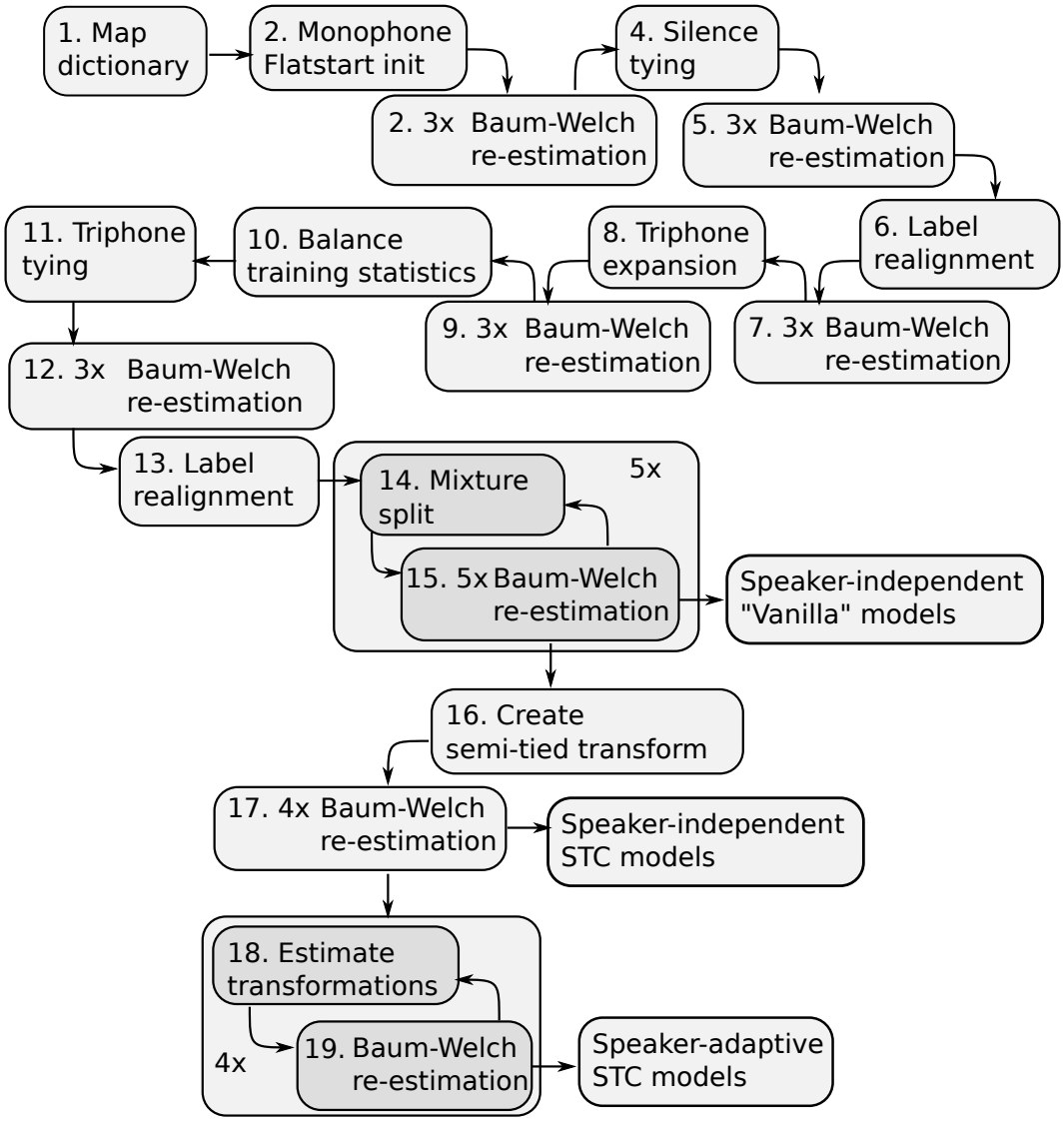


Figure 7.3: Training procedure of the baseline and test systems.

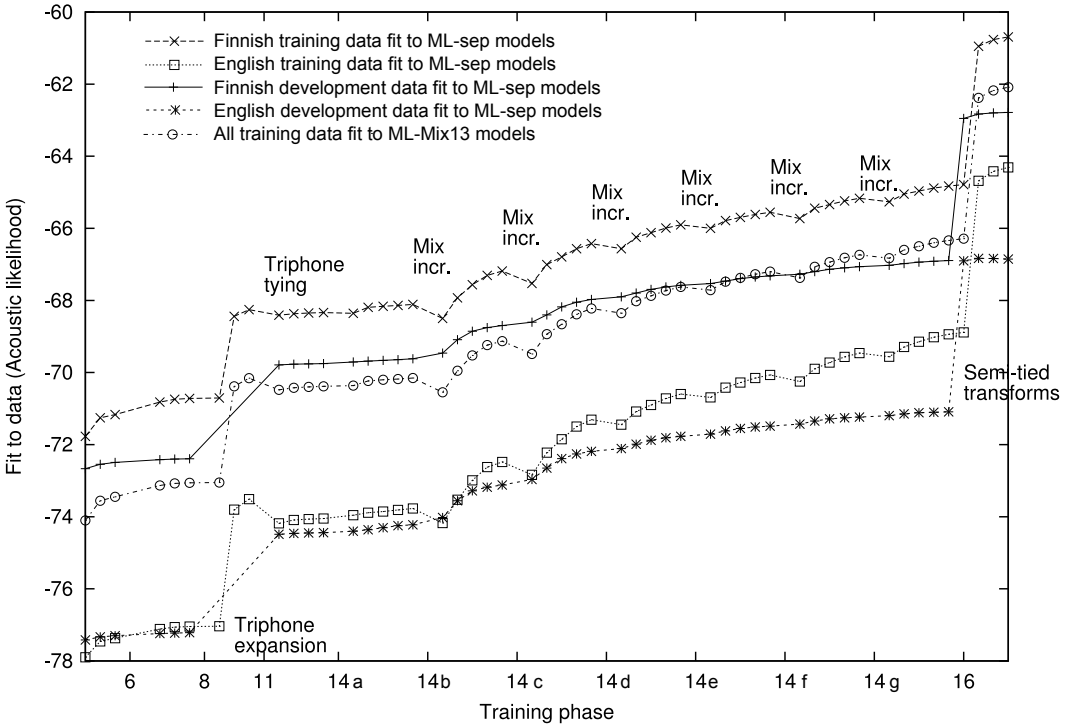


Figure 7.4: Likelihood increase in training. Note that likelihood is heavily dependent on the model geography, model set size and the data, so the likelihood is really only comparable with systems of identical model framework and similar data.
The model geometry changes on phases corresponding to training procedure presented in Section 7.8 are marked on the picture: Monophones are expanded to triphones at 8, the triphone models are tied at 11, mixtures are incremented on rounds at 14a-g, and finally and semi-tied transformations are applied at phase 16.

is referred to the aforementioned piece of HTK documentation.

The different steps in the training procedure are described very shortly below. Additional details have been added where the system is not consistent with standard training methods. Figure 7.3 illustrates the training procedure and Figure 7.4 shows the the likelihood changes in each step.

The fixed parameters used by all of the systems are listed in table 7.4.

1. **Create monophone labels with default pronunciations**.

   This is done separately for each source language in multilingual systems. In mixed systems that share data, the pooling of phones is defined first.

   The training transcriptions are derived from the word-level transliterations of the prompts. Whenever there are multiple pronunciations for a given word, the first one in the dictionary is used.

2. **Flat start monophones models** A prototype model is created from average properties of the signal with the HTK tool **HCompV**, using 10% of the available data. Variance floor is set to global 0.05 (absolute).

3. **Retrain models**

   In every training round, a new set of models is generated by embedded model re-estimation using the Baum-Welch algorithm implemented in the **HERest** tool. This kind of training does not require an accurately timed phone segmentation, instead the change from one phone to another is computed by the backward/forward probabilities. Every round changes the model set slightly, and several rounds of retraining - from 3 to 5 - is necessary to reach an adequate model set whenever the geometry of the models is changed.

4. **Create short silence** and tie into long silence middle state

   The short silence model is a *tee model*, that does not require a physical manifestation. Whenever it appears, it is identical to the centre state of the long silence model. The middle of the long silence model and the only state of the short silence are tied together, so that they are tried together whenever one of them appears. All the model geometry changes are done using the **HHEd** tool.

5. **Retrain models**

6. **Realign labels**

As mentioned earlier, the initial training transcriptions do not take into account different ways of pronouncing words. Whenever realigning the training labels, the acoustically best fitting pronunciation of available pronunciations is used in phonetically transcribing each word.

7. **Retrain models**

8. **Copy the monophone models to context-dependent triphone models**: Create some thousands of triphone models from a few dozen monophone models. Tie transitions of all triphones with same middle states to a single transition.

9. **Retrain models**

10. **Balance statistics across corpora for decision tree tying**

    This is actually only for future use in situations where the source corpora are not of roughly equal size. So we do nothing here for now.

11. **Tie triphone models**

    A phonetic decision tree is used to combine models that do not have adequate training data and to synthesise previously unseen models.

    For the mixed systems this tree is created by combining the phonetic questions over languages, based on phonetic classifications.

    The **HHEd** commands RO, QS and TB are used to build the trees and perform the tying. The AU command is used to synthesise previously unseen models and CO command to compact the list of HMMs by merging duplicates.

12. **Retrain models**

13. **Realign labels**

14. **Increase the number of Gaussians**

    The Gaussian count of a model is increased by splitting one or several of the Gaussians with the highest weight, and moving one Gaussian by 0.2 standard deviations up and the other down. Because of the splitting procedure, care must be taken not to increase the number of Gaussians by too many each round.

    The **HHEd** command MI is used to increase

15. **Retrain models**

    Repeat the previous step and this one until a desired number of Gaussians is reached.

    The question when to stop training is a very relevant one, and no definite answer can be given in this work. Generally, there is a desire to have enough Gaussians to give good coverage to the variations in the training and test data, but not too many to "over-fit" the training data, thus inhibiting model generality. Also, the number of Gaussians affects the ideal variance flooring values, which has a large impact on generality of the models.

    Also, more complex Gaussian mixtures require more training data - As a rule of thumb, ate least 200 frames of data should be used per Gaussian.

    Instead of digging into these issues, mixtures of 16 Gaussians are used.

16. **Create Semi-tied Covariance (STC) transforms**

    A semi-tied transform is created as an input transform for each triphone centre phone, so transforms *-/a/+* is applied to all triphones with /a/ as their centre phone, *-/b/+* to all triphones with /b/ as their centre phone and so on.

    The plain system has a diagonal covariance matrix. The semi-tied transform rotates the model's covariance vector so that the resulting covariance matrix describes the the variations in the training data better. Thus, a high increase in acoustic likelihood is inevitable, even though the real performance of the system does not increase as dramatically.

    The centre-phone specific transform is quite robust as there is a substantial amount of data for the training of each transform.

    Some of the systems cannot use a phone-specific transform. These systems use a single covariance transform. Additionally, tied-mixture systems cannot use a STC transforms because of the limitations in the training software.

17. **Retrain STC models**

    These training rounds use the newly created transforms.

18. **Create transformations for speaker-adaptive training (SAT)**

    A regression tree is created for the models and a set of CMLLR transforms is created for each speaker in the training data set.

19. **Retrain STC models with SAT**

    The last rounds of training are done using the Semi-tied input transform and the CMLLR transforms for each speaker in order to try to find a model set more independent of speaker variation.

7.9 Multilingual system training

Following the approaches introduced in Schultz & Kirchhoff, 2006, multilingual systems of three different types were built. The performance of these systems is evaluated in the next chapter.

The goal was to create a unified training procedure that can be applied to any language combination by simply including new items into the lists of training waveforms, training prompts, phonetic question pool and pronunciation dictionary.

Before we delve into these systems, a few words must be written regarding the risks of blindly combining data.

7.9.1 Sharing acoustic data between corpora

When using several corpora to teach a recognition system, or using a recogniser trained with one corpus to recognise utterances from another corpus, we have to spare a thought to the similarity of the corpora and what it means to the evaluation of the results.

The recording conditions of the corpora vary more or less. Most probably a different microphone is used to record the sentences, a different sound card is used to do the AD-transform, a different program is used to post-process the utterances.

In order to proceed with the work, we have to do a little injustice to the finer aspects of acoustics and make some very strong assumptions that stand on very dubious ground.

Firstly, we'll to presume that the corpora are acoustically similar enough. The inevitable differences in microphone, background noise and sound post-processing are essentially eliminated by using various normalisation techniques - DC removal (ZMEANSOURCE = true) , energy normalisation (ENORMALISE = true) and cepstral mean subtraction (TARGETKIND = MFCC_Z).

Secondly, we will crudely assume that the phone sets are ideal for the languages they represent. That means that we should not combine phones within language. Therefore it is better to leave the phone sets as they are, and only try to find similarities between them, rather than spend the time trying to find a better phone

set, that would be better at covering several languages, even with the price of performing more poorly on a single language case. When making this assumption, we are actively trying to forget the results of Dines *et al.* , 2009, showing that there are phone sets and dictionaries that yield better results for recognising English.

We will now go through the various methods of combining acoustic models in more detail.

ML-sep: Baseline combination

The ML-sep system is quite a simple one, as far as multilingualism is considered. The system consists of a combination of separate language models. The bilingual ML-sep system in this work is thus a copy of the baseline systems, and works as a baseline for comparing multilingual performance.

ML-mix: Phone combination

Here the training data was put together and was used to train systems. A varying amount of phones were combined across languages, according to the KL distances (see Chapter 5.3.5) of single-Gaussian models of the baseline recognisers.

Three different methods were used to combine training data for the models.

ML-mix 0 to 13 In the systems ML-mix$_0$, ML-mix$_3$, ML-mix$_6$, ML-mix$_9$ and ML-mix$_{13}$ monophones are combined across languages. The distance between two phone models was calculated by adding together the distances of the emitting Gaussians of each respective HMM state of the two models. Single-Gaussian models were used here. A graphical representation of the monophone distances is shown in Figure 7.5.

In each of the ML-mix$_n$ systems, n closest phone pairs were marked as the same phone across languages. A straightforward mapping was available for 13 phone pairs, as listed in table 7.5. Combining more phones in a simple way is difficult, as no more a straightforward 1-to-1 mappings exist. The selection of English phones is larger than that of the Finnish ones, and for example, the English vowels /ay/, /aw/ and /ao/ are closer to each other than any of the Finnish vowels.

Triphone expansion is performed without further considerations about the language context. Triphones are then tied using a shared decision tree trained using data from both languages. Phonetic knowledge of both languages is used to group questions of both languages into a compatible format, combining the questions where appropriate. Note that the number of triphones in the training data that are truly shared across languages is less than 20% even for the ML-mix$_{13}$ system.

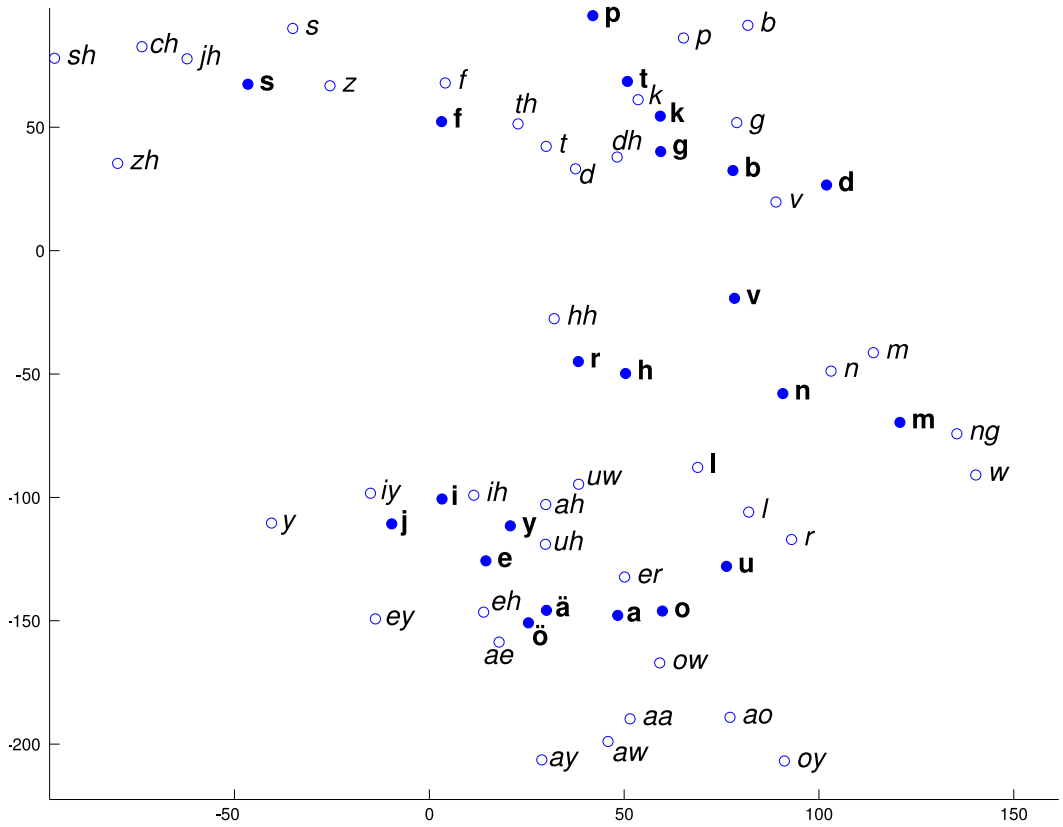


Figure 7.5: Multi-factor dimensionality reduction (MDR) graph of phones of WSJ and Speecon corpora. Finnish phones (Speecon phone set) are marked with solid circles and labelled with bold type text, English phones (CMU phone set) are displayed as circle outlines and labelled with italic type text.

English	Finnish	Distance
f	f	3.75
n	n	4.64
g	g	4.83
eh	e	4.87
k	k	4.95
ow	o	5.92
m	m	6.90
iy	i	6.98
z	s	9.84
hh	h	10.47
y	j	14.92
d	d	16.08
aa	a	8.15

Table 7.5: Closest monophone pairs between the baseline recognisers. The distance is a sum of Kullback-Leibler divergence over all states of the respective HMM models. The average distance of phones calculated this way is 67.98 with a standard deviation of 41.53, so a distance under 5 should be considered a very close match. For the last pair, even though acoustic similarity is apparent, the mapping is not straightforward as /a$_{fi}$/ is also quite similar to /aw$_{en}$/

The number of semi-tied covariance transforms however unifies the phone models across languages for the STC-model sets.

ML-mix 22 Additionally, ML-mix$_{22}$ system is built based on knowledge of the phone sets and the IPA chart. This system is contrary to our second assumption: Here the English diphtongs have been broken down into two separate phones so that their vowel slide is represented by context-dependence of the models. Thus, based on phonetic knowledge, we can reach a system that has 38 phones, of which 1 is unique to Finnish and 14 are unique to English.

The phone set of this system, which can be at best described as a curiousity, is shown in table 7.7.

System	Combination technique	Mono-phones	States Total	Mixed	Fin	Eng	Cov. xforms
Baseline Fin	-	23+2	2405	0%	0%	100%	24
Baseline Eng	-	39+2	2244	0%	100%	0%	40
ML-sep	Separate models	62+4	4649	0%	51.7%	48.3%	64
ML-mix$_0$	Monoph. KLd	62+2	4652	0.1%	51.5%	48.5%	63
ML-mix$_3$	Monoph. KLd	59+2	4659	2.7%	50.2%	47.2%	60
ML-mix$_6$	Monoph. KLd	56+2	4654	6.3%	47.6%	46.1%	57
ML-mix$_9$	Monoph. KLd	53+2	4661	14.2%	40.9%	44.8%	54
ML-mix$_{13}$	Monoph. KLd	49+2	4660	22.9%	35.4%	41.8%	50
ML-mix$_{22}$	Monoph. Rules	32+2	4631	65.0%	13.5%	21.4%	33
ML-mix$_{100}$	Tied state KLd	62+2	4556	2.3%	50.7%	47.0%	1
ML-mix$_{200}$	Tied state KLd	62+2	4462	4.5%	49.7%	45.8%	1
ML-mix$_{577}$	Tied state KLd	62+2	4181	13.7%	46.3%	40.0%	1
ML-tag	Full tying	62+2	4646	-	-	-	0

Table 7.6: Phone and Gaussian counts of baseline and test systems. First number in monophone field is the phone model count, the second silence model count. The ML-tag system is a tied-mixture system, and has a pool of Gaussians, and the phone models are list of weights of which of the pooled Gaussians should be used.

ML-mix 100 to 577 In the systems ML-mix$_{100}$, ML-mix$_{200}$ and ML-mix$_{577}$ triphone states are combined across languages. The reason for using a sub-phone level is the state transition geometry. Transitions are tied across phones (training procedure part 11) and by merging complete triphones across languages, it would be necessary to either create a new transition - when there might be little training data - or to crudely say, that both models use a transition from one language only.

The distance is calculated between the single Gaussian tied-triphone models of the baseline recognisers. The 2244 tied states of the WSJ0 baseline and the 2405 states of the Speecon baseline give around 5 million distance pairs. Of these, there are 577 pairs where the mapping is symmetrical. This means that these phones are the closest neighbours for each other, as opposed to there being a "chain" of closest neighbours. Figure 7.6 tries to illustrate this.

In each of the ML-mix$_n$ systems, n closest states are marked as the same. The

Phoneme	English example and translation		Finnish example and translation	
A	odd	A d d	rotta	r o t t A
ae	at	ae t	mäyrä	m ae y r ae
a	cow	k a u		
ah	hut	h ah t		
b	be	b i	bertta	b e r t t A
ch	cheese	ch i i z		
d	dee	d i	daavid	d A A v I d
dh	thee	dh i i		
E	Ed	E d		
e	ate	e I t	eemeli	e e m e l i
f	fee	f i	faarao	f A A r A o
g	green	g r i n	gaselli	g A s e l l I
h	he	h i	herne	h e r n e
i	eat	i t		
I	it	I t	paavi	p A A v I
jh	gee	jh i i		
k	key	k i	kala	k A l A
l	lee	l i	kala	k A l A
m	me	m i	mäyrä	m ae y r ae
n	knee	n i	noppa	n o p p A
ng	ping	p i ng	aurinko	A u r I ng k o
o	odd	o d	rotta	r o t t A
oe	hurt	h oe t	yö	y oe
O	ought	O t		
p	pee	p i	paavi	p A A v I
r	read	r i i d	rotta	r o t t A
s	sea	s i	sää	s ae ae
sh	she	sh i i		
t	tea	t i	rotta	r o t t A
th	theta	th e i t ah		
u	two	t u		
U	hood	h U d	ukko	U k k o
v	vee	v i	paavi	p A A v I
w	we	w i		
j	yield	j i l d	jouhi	j o U h I
z	zee	z i		
zh	seizure	s i i zh oe		
y			mäyrä	m ae y r ae

Table 7.7: A proposal for a "Finnified" combined phoneme set, where the modelling of diphthongs as single phones has been replaced with context-dependency. This set is used in the ML-mix$_{22}$ system.

distance thresholds are 4.6, 5.7 and 15.3 for the $n = 100$, $n = 200$ and $n = 577$ systems respectively. The average distance of all phone pairs is around 78 with a standard deviation of about 40.

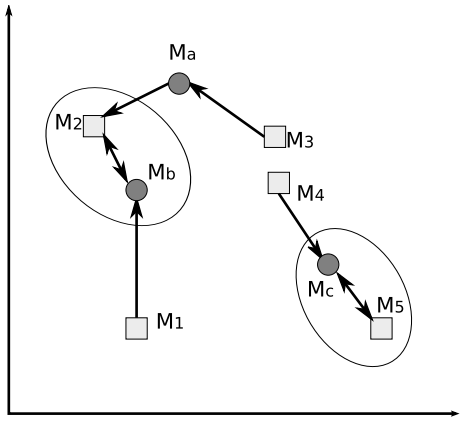


Figure 7.6: Exemplary distances between L_1 acoustic models M_1 - M_5 and L_2 acoustic models M_a - M_c drawn into a two-dimensional graph. Closest cross-lingual mapping for each phone is shown with an arrow. Only M_b-M_2 and M_c-M_5 make symmetrical pairs, as they are closest to each other. M_3 or M_4 cannot be combined to L_2 models, as these are paired to other L_1 models. M_3 and M_4 could make a fine pair, but the tweaking of intra-lingual tying remains outside the scope of this work.

ML-tag: Fully tied baseline recogniser

The final system built for this work is a tied-mixture system applying the ML-tag parameter sharing scheme. The basis for this system consists of the two baseline recognisers. During training, a copy of the recognisers is made at 12th training step, before incrementing the number Gaussian mixtures. These tied, single Gaussian systems are combined and their Gaussians are pooled to create a tied-mixture system. These models are trained for 8 rounds, after which the training procedure is again continued from step 16.

Thus, the resulting recogniser has a unique model for every phone but all the models share the same pool of Gaussians. Thus, adaptation of the models of one language will automatically mean the adaptation of all models using the same Gaussians.

7.10 Analysis of multilinguality

Tables 7.6 and 7.8 contain some numbers relating to acoustic data sharing across the training corpora for various combintaion approaches.

The state sharing listed in table 7.6 has been generated by making lists of states used in the training corpora for both languages, and seeing how many states overlap.

The analysis of triphone tying context question usage in table 7.8 reveals how

System		Total	Mixed	Multiple	Finnish	English
Base Fin	Different questions	190	0.0%	0.0%	100.0%	0.0%
	Questions used	2682	0.0%	0.0%	100.0%	0.0%
	Deciding questions	1780	0.0%	0.0%	100.0%	0.0%
Base Eng	Different questions	212	0.0%	0.0%	0.0%	100.0%
	Questions used	2661	0.0%	0.0%	0.0%	100.0%
	Deciding questions	1816	0.0%	0.0%	0.0%	100.0%
Sep	Different questions	402	0.0%	0.0%	47.3%	42.7%
	Questions used	5343	0.0%	0.0%	50.2.0%	49.8%
	Deciding questions	3596	0.0%	0.0%	49.5%	50.5%
Mix-0	Different questions	280	0.0%	43.6%	24.3%	32.1%
	Questions used	5357	0.0%	64.9%	18.8%	16.4%
	Deciding questions	3608	0.0%	59.7%	21.0%	19.3%
Mix-3	Different questions	280	27.1%	22.1%	20.7%	30.0%
	Questions used	8042	36.7%	41.3%	11.8%	10.2%
	Deciding questions	3644	5.3%	57.4%	19.5%	17.9%
Mix-6	Different questions	280	45.0%	10.7%	17.1%	27.1%
	Questions used	8820	46.5%	34.3%	10.7%	8.5%
	Deciding questions	3673	12.3%	51.8%	19.9%	16.0%
Mix-9	Different questions	280	53.6%	7.1%	14.3%	25.0%
	Questions used	9407	56.3%	28.7%	7.4%	7.6%
	Deciding questions	3728	23.9%	46.3%	14.6%	15.2%
Mix-13	Different questions	280	65.7%	2.1%	10.7%	21.4%
	Questions used	9879	65.7%	22.0%	5.5%	6.8%
	Deciding questions	3763	36.8%	37.4%	11.4%	14.4%
Mix-22	Different questions	268	86.6%	0.0%	0.7%	12.7%
	Questions used	10732	83.5%	11.5%	0.4%	4.6%
	Deciding questions	3844	66.4%	22.1%	1.1%	10.5%

Table 7.8: Phonetic question usage in triphone tying for various systems. Mixed means questions that contain (possibly among other) models that are shared across languages (e.g. +/a/,+/ah$_{en}$/); Multiple means questions that contain questions of both Finnish and English (e.g. +/a$_{fi}$/,+/a$_{en}$/); Finnish and English mean exclusively intra-lingual questions. The middle phone is counted as a question in this analysis.

often shared contexts are used in phonetic decision trees (see Section 3.6.3). The questions are used only as an aid for clustering. The real magic is in the euclidean distance. When a cluster is split, the splitting is done based on the question that gives the most fitting division for the two new nodes. When a single cluster can no longer be divided without losing information, the cluster is considered to be a leaf

	Vanilla models		STC models	
Task	1st pass	Adapted	1st pass	Adapted
Speecon devel	-	-	3.3	-
Speecon eval	4.2	3.4	3.7	3.5
EMIME finnish	4.2	2.4	3.4	2.7
WSJ devel (w1 h1 p0)	-	-	8.7	-
WSJ eval (w0 si et 05)	5.6	3.7	4.8	4.3
EMIME English	41.6	29.6	36.8	31.5

Table 7.9: Baseline performance on English development and evaluation tasks. Error measurement is WER for WSJ and EMIME English data sets, and LER for Speecon and EMIME Finnish data sets.

node in the clustering tree, and the question that was used to generate the leaf node is referred to as a deciding question.

From the high percentage of mixed deciding questions we can see that corpus specifity is not as deciding as initially feared. Even though mixed corpora questions are slightly underrepresented in the "decisive" questions leading to the leaf nodes of the tree, we can clearly see that the usage of mixed questions is high and that triphone tying across corpora works better and better as more phones are defined to be the same across corpora.

7.11 Baseline tests and results

A batch of baseline recognisers with different triphone tying parameters was first trained and tested with the development sets. The best combination of these parameters are shown in Table 7.4. The baseline recognisers were then further optimised by tweaking the language model weight again using the development set. Finally the evaluation sets were recognised with these baseline recognisers.

All tests were made in an identical fashion. First, word lattices are generated with the **HDecode** decoder using a bigram language model, an average beam width (250) and a loose restrain on the decoder stack size (30000).

These lattices were then expanded and rescored using **SRILM lattice-tool** and a higher order ngram. The first round recognition result is the best path through this lattice.

Tests using adaptation were made by generating the adaptation transforms using the output from the first round. First, the **HVite** decoder was used to align the first pass labels onto the data. This label data had been lost in the lattice expansion phase. Then, **HHEd** is used to create a regression tree of transforms. **HERest** is used to train the transformations.

Then a new batch of lattices is created with the **HDecode** decoder using the new transformations. The second pass final result is again derived by rescoring the adapted lattices with an n-gram of a higher order.

The recognition time varies depending on the acoustic models set, ranging from around 10-15 real-time for lattice generation and less than 0.5 RT for the remaining operations. The baseline performance results are shown in table 7.9.

For the tied-mixture system, a less optimised Viterbi decoder **HVite** is used. The beam had to be set to a low value (180) to constrain decoding times. Due to the geometry of the models, adaptation did not work quite as well as hoped.

Chapter 8

Tests results

In this chapter we will go through the tests made on the systems that were developed as described in the previous chapter.

The tests were done in two parts:

1. Performance on recognising Finnish and English sentences. This also includes tests about intra-lingual adaptation.

2. Performance in cross-lingual adaptation. Here a set of sentences from language A are used to adapt the recogniser in order to improve recognition performance for sentences of the same speaker in language B.

8.1 Test setups and result representation

Testing speech recognisers is very straightforward. A test consists of running the speech recogniser with predefined parameters on a predefined set of utterances, *the evaluation set*. The parameters that control the recogniser, for example the beam width or the stack size, are usually optimised by running the recogniser on a different set of utterances, *the development set*.

For speaker-independent, large-vocabulary recognisers, where the test set includes complete sentences from several speakers, the average word error rate (WER) of either all the sentences or all the speakers is given.

The scoring program used in this thesis is *sclite*[1], which aligns the recognition results to the reference labels and gives the amount of errors as well as an average of error rates of all sentences and the average of the average error rates of all speakers. These error rates are reported with an accuracy of one decimal.

[1]From the NIST scoring package, see http://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sctk.htm

Beside the graphs showing absolute differences in errors, I have included charts that show statistically significant differences between systems. The statistical test is the matched pair test (see Section 4.2.2) done with the program built for the purpose, *sc_stats*[2]. In the charts each element represents a test between the system indicated by the column and the system indicated by the row. The system that performs significantly better is indicated by an arrow. These charts are included in appendix B.

8.2 Tests on combined acoustic models

These tests confirm that all recognisers perform adequately. The results are shown in Figure 8.1 and in appendix A. Selected statistical significance tests are shown in tables B.1-B.8 in appendix B.

8.2.1 Test setup

As described in the earlier chapter, a group of recognisers was trained.

First, a combination of two separate system acts as the baseline. This is known as ML-sep recogniser.

Secondly, a group of ML-mix recognisers was trained. These combine the training data across languages. These range from mix_0, where the data is just pooled, to mix_{22}, where 22 phones from both languages are marked as the same phone. Variants of ML-mix with various mixing strategies and thresholds.

Furthermore, a bilingual tied-mixture system was built, and is known as ML-tag.

A basic set of models and a more advances model set with phone-specific semi-tied covariance transform is trained for each of these recognisers. The first set is called "vanilla", the second "STC". This yields a set of 16 recognisers. Three tests are done on each of these:

- Single-pass recognition,
- Two-pass recognition; second pass with a single CMLLR transform and
- Two-pass recognition; second pass with a regression tree CMLLR transform.

[2]Also from the aforementioned NIST scoring package.

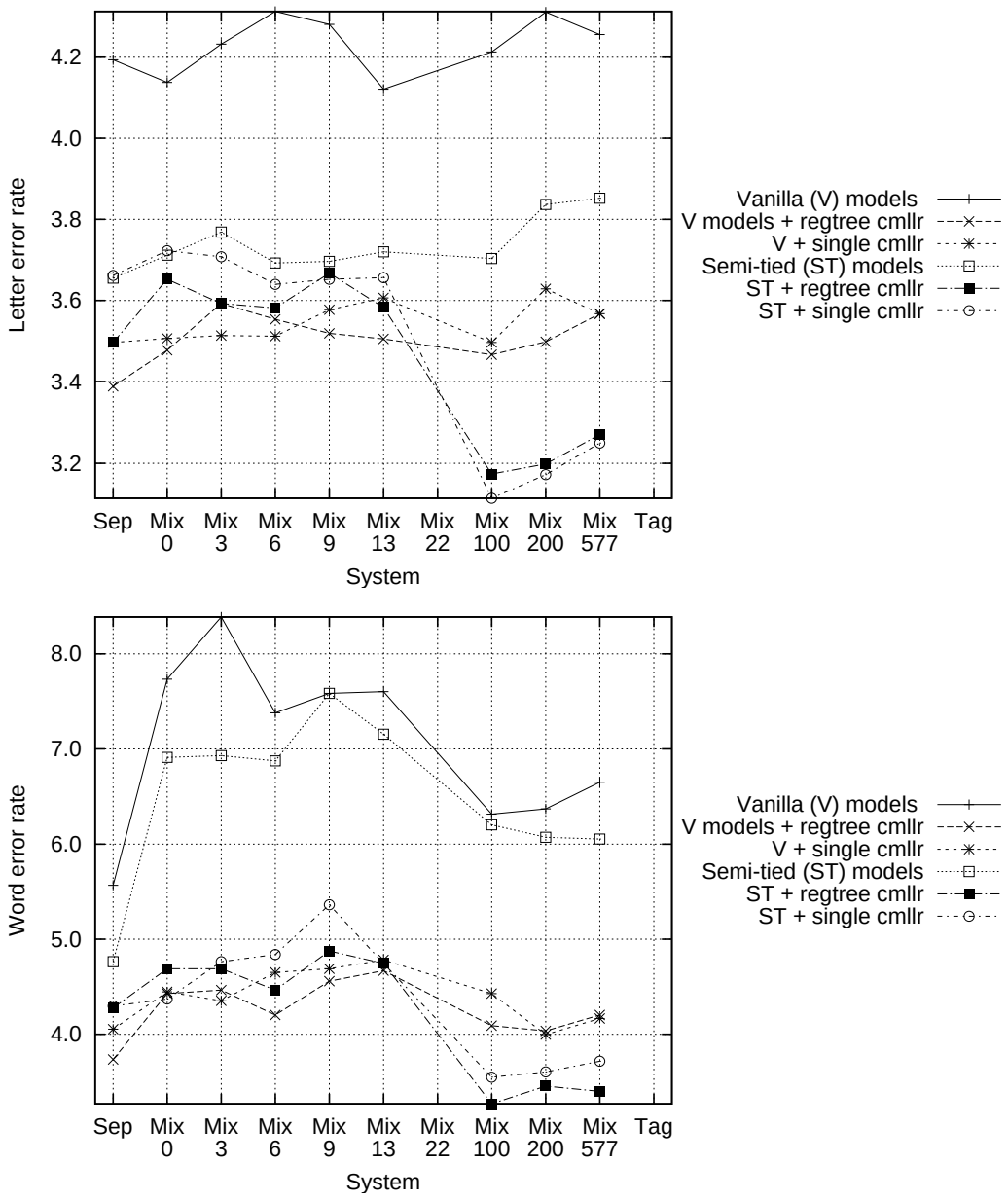


Figure 8.1: Recognition results of Speecon and WSJ data sets. ML-mix$_{22}$ and ML-Tag results are so bad that they have been removed from the figure. They can be found in appendix A.

8.3 Cross-lingual adaptation tests

An interesting question with multilingual acoustic modelling is naturally the possibility of using adaptation data over languages. The tests are done using the bilingual EMIME utterance collection. The results are shown in figures 8.2 and 8.3 and in appendix A. The significant differences in systems are shown in tables B.9-B.12 in appendix B.

8.3.1 Test setup

The system description is similar to the earlier tests on adapted systems. The array of tests for each recognisers is a bit wider. There are sentences in two languages for each speaker in the corpus. When we want to recognise data in language A, we can try to use adaptation data generated either from language A or language B.

The list of tests is as follows:

- Single-pass recognition on language A,
- Two-pass recognition; 1st pass on A, 2nd pass with a single CMLLR transform on A,
- Two-pass recognition; 1st pass on A, 2nd pass with a regression tree CMLLR transform on A.
- Cross-adapt; 1st pass on B, 2nd pass with a single CMLLR transform on A
- Cross-adapt; 1st pass on B, 2nd pass with a regression tree CMLLR transform on A
- Phoneloop; 1st pass on B with a phoneme recogniser, 2nd pass with a single CMLLR transform on A
- Phoneloop; 1st pass on B with a phoneme recogniser, 2nd pass with a regression tree CMLLR transform on A

The last tests involve phoneme recognisers, as introduce in section 3.5.2. These tests are included as references to see whether there is any advantage in making the effort of combining the acoustic models across languages.

Note that the ML-sep system cannot do cross-lingual adaptation.

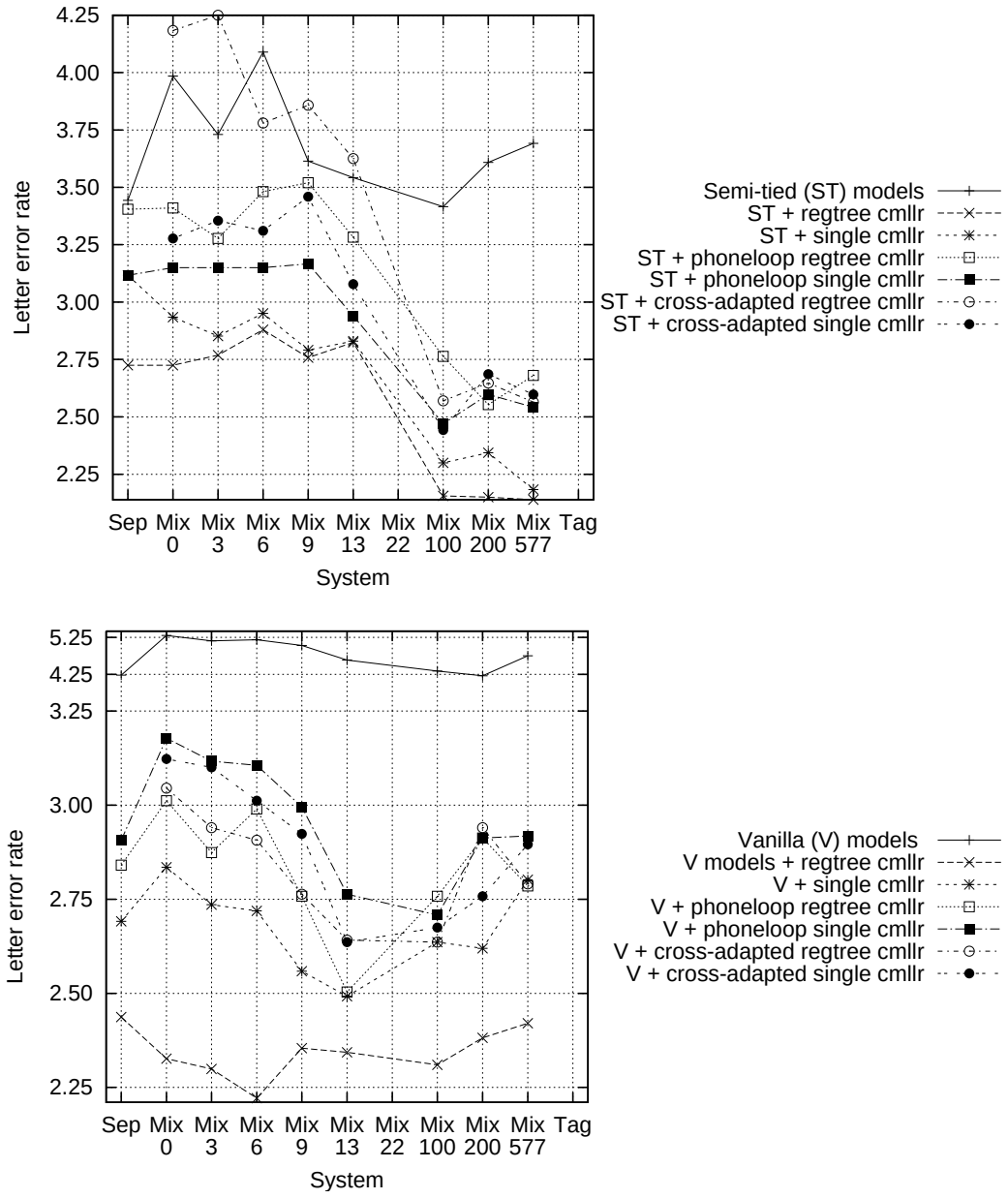


Figure 8.2: EMIME data recognition results for Finnish data set. ML-mix$_{22}$ and ML-Tag results are so bad that they have been removed from the figure. They can be found in appendix A.

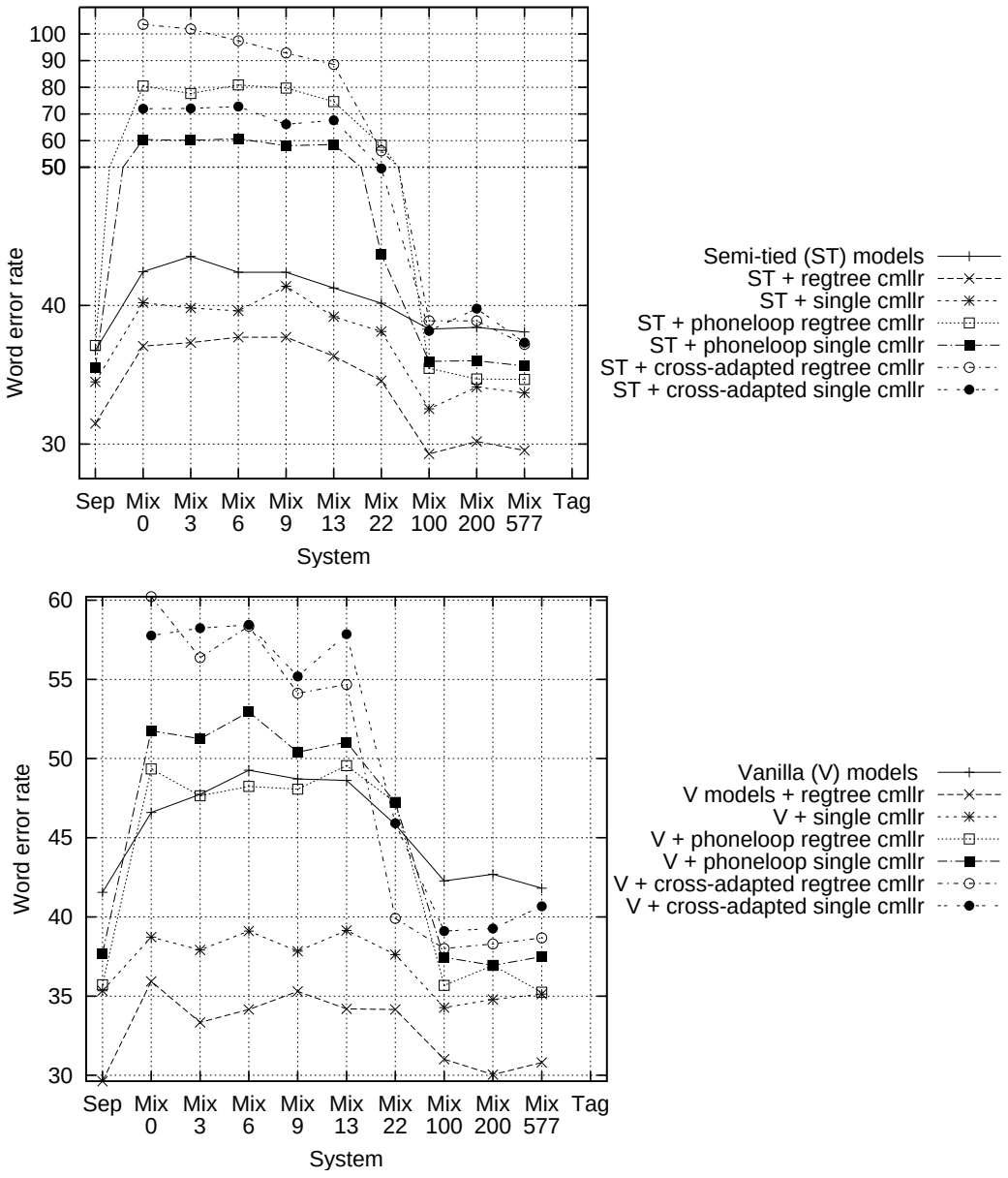


Figure 8.3: EMIME data recognition and adaptation results for English set. ML-Tag results are so bad that they have been removed from the figure. They can be found in appendix A.

8.4 Result analysis

First, we notice that ML-mix$_{22}$ and ML-Tag systems do not perform adequately - With ML-mix$_{22}$ this is most likely due to the disproportionate reduction of the phone set. With ML-Tag the reasons can be an improper training procedure or problems in the software toolkit regarding this kind of model set - or both.

The next obvious observation is that the recognition of English sentences spoken by Finnish speakers is quite difficult. The error difference is comparable to the error increase when American and British English recognisers are used across dialects, as investigated by Yan & Vaseghi, 2002. Here, the American sentences recognised with an American recogniser got 8,8 WER, whereas British sentences recognised by the same yielded 30.6 WER. Therefore the error of around 35 WER for the Finnish English sentences does not seem so bad, especially as the EMIME sentences have a much larger vocabulary and thus a more confusion-prone language model is used.

In their work, Schultz & Waibel, 1998 managed to contract the amount of parameters for a 5-language recogniser by 60% with only a slight reduction in recognition performance - An average of 20% reduction per additional language. The merging of two languages in this work with a 13 % reduction in parameters and no reduction in accuracy and 22% reduction with slight performance degradation suggests that the Finnish-English language pair is at least as similar as the set of five languages Croatian, Japanese, Korean, Turkish and Spanish.

The comparison of adaptation methods with various recognisers gives us a clear winner: The ML-mix$_{100}$-system. Unfortunately, its performace edge was not predicted. The ML-mix$_{100}$, ML-mix$_{200}$ and ML-mix$_{577}$ have a different covariance transform applied to them, and hypothesis was originally, that therefore they would perform worse. In fact these three systems did a better job in adaptation that any of the other systems and therefore a straight comparison to other systems might not be fair.

However, it is apparent that none of the truly cross-lingual adaptation attempts with similar STC transform geometry do significantly better than the baseline recogniser with its phoneme loop adaptation procedure. This probably has to do more or less with the fact that the ML-sep system's unadapted performance is unchallenged by any of the ML-mix systems.

Looking back at table 7.6, it is apparent that the multi-lingual tying does as it is supposed to do with ML-mix$_0$: The Gaussian count is identical to the ML-sep system, as it should be as theoretically nothing has changed. An educated guess is that the clustering parameters for triphone tying should be tweaked individually

for each ML-mix system. There might also be something inherently wrong with the idea of combining phonetic questions across languages, or maybe some flaws in their grouping.

Chapter 9

Conclusions

From the results presented in the previous chapter we can conclude the following:

- Recognition of native speech (Finnish) can be enhanced with adaptation data in foreign language (English), whereas recognition of foreign speech (English) cannot be helped with the adaptation data from native language (Finnish).
- Up to 13 phones may be combined across languages without affecting the recognition of the native language (Finnish). This is actually the maximum of straightforward combinations possible using the Kullback-Leibler distance metric. Further combinations would require a one-to-many-mapping of phones across languages. The system with a very aggressive knowledge-based merging strategy, ML-mix$_{22}$, degrades the performance significantly.
- Overall the best adaptation strategy is to use the baseline recogniser for intralingual adaptation label generation and ML-mix$_{100}$-STC system for 2nd pass. For cross-lingual adaptation, ML-mix$_{100}$-STC and a phoneme loop should be used to create adaptation matrices and the same system should be used for the 2nd pass.

There are certain issues that remain to be investigated.

- **Generality:** The tests were run on the language pair Native Finnish vs. Foreign English. Are similar results obtainable from other language pairs? What about including other languages? Test sets were also very small. Are similar results obtainable with some dozens of speakers? Or hundreds?
- **Corpus size balancing:** The amount of speech in the English training corpus is almost thrice the amount of speech in the Finnish training corpus. The

English phone set was around 30% larger than the Finnish set, and the training set includes 15431 different triphones, whereas the Finnish set only 5806 - Thus, average data per triphone is almost equal. A more robust recogniser could be trained by using the WSJ1 and Speecon corpora, but the amount of English data would overwhelm the Finnish data. How should this be resolved? One possibility would be to use some kind of information criteria to control the clustering procedure as presented in Lyu & Lyu, 2008.

- **Phone set optimisation:** The phone sets of the recognisers were taken as granted. A further investigation in the properties of individual languages and the unifications of phone sets of language pairs might reduce the error further.

  An alternative is a one-to-many-mapping of phones when combining languages, where a single phone of a language that has a sparse phone set represents several phones in a language with a more diverse phone set.

  Furthermore, it has been shown that for the English language some pronunciation dictionaries with a larger phone set and restricted distribution perform better than the free CMU dictionary used in this work (Dines *et al.* , 2009).

  From table 7.6 we notice, that monophone pooling together with shared phonetic question trees does not automatically contract the model set.

- **Role of the covariance transform geometry:** The advantages of ML-mix$_{100}$ might come from its different covariance transformation geometry. If this is the case, it is hardly motivated to create mixed systems, as the the ML-sep recogniser using a simple phoneme recogniser to create adaptation transcriptions can do as well as the cross-adapted multilingual recognisers.

- **Approach to acoustic modeling and distance metrics:** A simple, 16-Gaussian mixture model was used for all the phone models. An alternate approach, that could somewhat solve the questions about corpora size, is to use a fully tied HMM system, better known as tied-mixture system. This has a separate pool of Gaussian mixtures (codebook), and the phone model is nothing more than a list of Gaussians which the phone uses and their assosiated weights.

  This approach would give us a very simple distance metric: The (squared) difference of codebook weights between phones.

- **Scrutinous investigation of test system:** Speech recognisers are complex systems, and a research system built for testing a particular problem can be

a prime example of patchwork, kludging and general inelegancy of software engineering. The bottom line is that there remains a huge domain for human errors, and a mistake made in the preliminary preparations can have a terrible effect on the end results.

Epilogue:

A declaration of disbelief in the foundations of the findings of this thesis

An important thing that deserves a mention in the end: Although the results are confirmed by statistical significance tests, the main test set consisted of only 3 speakers, whose voices are also somewhat similar.

So I advice the reader to remain sceptical as ever about the results reported in this thesis.

Appendix A

Test result error summary

The tables on the following pages show all the results of the tests done with the recognisers trained for this work.

	System		1st pass	Intra-lingual CMLLR single	Intra-lingual CMLLR regtree	Cross-lingual CMLLR regtree	Cross-lingual CMLLR single	Phoneloop CMLLR regtree	Phoneloop CMLLR single
wsj0 si_et_05	Sep	Vanilla	5.6	4.1	3.7	-	-	-	-
	Sep	STC	4.8	4.3	4.3	-	-	-	-
	Mix- 0	Vanilla	7.7	4.4	4.4	-	-	-	-
	Mix- 0	STC	6.9	4.4	4.4	-	-	-	-
	Mix- 3	Vanilla	8.4	4.4	4.5	-	-	-	-
	Mix- 3	STC	6.9	4.8	4.7	-	-	-	-
	Mix- 6	Vanilla	7.4	4.7	4.2	-	-	-	-
	Mix- 6	STC	6.9	4.8	4.5	-	-	-	-
	Mix- 9	Vanilla	7.6	4.7	4.6	-	-	-	-
	Mix- 9	STC	7.6	5.4	4.9	-	-	-	-
	Mix- 13	Vanilla	7.6	4.8	4.7	-	-	-	-
	Mix- 13	STC	7.2	5.1	4.7	-	-	-	-
	Mix- 22	Vanilla	54	41.9	38.4	-	-	-	-
	Mix- 22	STC	49.7	40.6	37.9	-	-	-	-
	Mix- 100	Vanilla	6.3	4.4	4.1	-	-	-	-
	Mix- 100	STC	6.2	3.5	3.3	-	-	-	-
	Mix- 200	Vanilla	6.4	4.0	4.0	-	-	-	-
	Mix- 200	STC	6.1	3.6	3.5	-	-	-	-
	Mix- 577	Vanilla	6.7	4.2	4.2	-	-	-	-
	Mix- 577	STC	6.1	3.4	3.7	-	-	-	-
	Tag	Vanilla	16.9	-	-	-	-	-	-

Table A.1: Recognition results from WSJ tests. Error is measured in WER.

	System		1st pass	Intra-lingual CMLLR single	Intra-lingual CMLLR regtree	Cross-lingual CMLLR regtree	Cross-lingual CMLLR single	Phoneloop CMLLR regtree	Phoneloop CMLLR single
EMIME English set	Sep	Vanilla	41.6	35.3	29.6	-	-	35.7	37.7
	Sep	STC	36.8	34.5	31.5	-	-	37.1	35.5
	Mix- 0	Vanilla	46.6	38.7	35.9	60.2	57.8	49.3	51.8
	Mix- 0	STC	42.4	40.2	37.1	102.5	71.8	74.4	60.4
	Mix- 3	Vanilla	47.7	37.9	33.3	56.4	58.2	47.7	51.2
	Mix- 3	STC	43.5	39.8	37.3	100.4	72.2	70.3	60.1
	Mix- 6	Vanilla	49.3	39.1	34.2	58.4	58.4	48.2	52.9
	Mix- 6	STC	42.4	39.6	37.7	97.4	72.7	72.2	60.7
	Mix- 9	Vanilla	48.7	37.8	35.3	54.1	55.2	48.1	50.4
	Mix- 9	STC	42.4	41.4	36.6	92.2	66.3	72.0	58.1
	Mix- 13	Vanilla	48.6	39.1	34.2	54.7	57.9	49.6	51.0
	Mix- 13	STC	41.3	39.2	36.4	88.6	67.6	66.4	58.6
	Mix- 22	Vanilla	45.8	37.6	34.2	39.9	45.9	43.7	47.2
	Mix- 22	STC	40.2	38.1	34.6	56.2	49.9	58.2	50.6
	Mix- 100	Vanilla	42.3	34.3	31.0	38.0	39.1	35.7	37.5
	Mix- 100	STC	38.3	32.5	29.3	38.9	38.2	35.5	36.0
	Mix- 200	Vanilla	42.7	34.8	30.0	38.3	39.3	34.4	36.9
	Mix- 200	STC	38.4	34.1	30.2	38.9	39.8	34.7	36.0
	Mix- 577	Vanilla	41.8	35.1	30.8	38.7	40.7	35.3	37.5
	Mix- 577	STC	38.1	33.7	29.5	37.2	37.3	34.7	35.6
	Tag	Vanilla	68.7	-	-	-	-	-	-

Table A.2: Recognition results from English EMIME tests. Error is measured in WER.

	System		1st pass	Intra-lingual CMLLR single	Intra-lingual CMLLR regtree	Cross-lingual CMLLR regtree	Cross-lingual CMLLR single	Phoneloop CMLLR regtree	Phoneloop CMLLR single
Speecon evaluation set	Sep	Vanilla	4.2	3.5	3.4	-	-	-	-
	Sep	STC	3.7	3.7	3.5	-	-	-	-
	Mix- 0	Vanilla	4.1	3.5	3.5	-	-	-	-
	Mix- 0	STC	3.7	3.7	3.7	-	-	-	-
	Mix- 3	Vanilla	4.2	3.5	3.6	-	-	-	-
	Mix- 3	STC	3.8	3.7	3.6	-	-	-	-
	Mix- 6	Vanilla	4.3	3.5	3.6	-	-	-	-
	Mix- 6	STC	3.7	3.6	3.6	-	-	-	-
	Mix- 9	Vanilla	4.3	3.6	3.5	-	-	-	-
	Mix- 9	STC	3.7	3.7	3.7	-	-	-	-
	Mix- 13	Vanilla	4.1	3.6	3.5	-	-	-	-
	Mix- 13	STC	3.7	3.7	3.6	-	-	-	-
	Mix- 22	Vanilla	38.4	37.5	37.3	-	-	-	-
	Mix- 22	STC	37.6	37.0	36.6	-	-	-	-
	Mix- 100	Vanilla	4.2	3.5	3.5	-	-	-	-
	Mix- 100	STC	3.7	3.1	3.2	-	-	-	-
	Mix- 200	Vanilla	4.3	3.6	3.5	-	-	-	-
	Mix- 200	STC	3.8	3.2	3.2	-	-	-	-
	Mix- 577	Vanilla	4.3	3.6	3.6	-	-	-	-
	Mix- 577	STC	3.9	3.2	3.3	-	-	-	-
	Tag	Vanilla	6.4	-	-	-	-	-	-

Table A.3: Recognition results from Speecon tests. Error is measured in LER.

	System		1st pass	Intra-lingual CMLLR single	Intra-lingual CMLLR regtree	Cross-lingual CMLLR regtree	Cross-lingual CMLLR single	Phoneloop CMLLR regtree	Phoneloop CMLLR single
EMIME Finnish set	Sep	Vanilla	4.2	2.7	2.4	2.8	2.9	2.8	2.9
	Sep	STC	3.4	3.1	2.7	3.9	3.2	3.4	3.1
	Mix- 0	Vanilla	4.0	2.9	2.7	4.2	3.3	3.3	3.2
	Mix- 0	STC	5.3	2.8	2.3	3.0	3.1	3.1	3.2
	Mix- 3	Vanilla	5.2	2.7	2.3	2.9	3.1	2.9	3.1
	Mix- 3	STC	3.8	2.8	2.6	4.2	3.4	3.3	3.1
	Mix- 6	Vanilla	5.2	2.7	2.2	2.9	3.0	3.0	3.1
	Mix- 6	STC	4.1	2.9	2.7	3.8	3.3	3.6	3.2
	Mix- 9	Vanilla	5.0	2.6	2.4	2.8	2.9	2.7	3.0
	Mix- 9	STC	3.8	2.8	2.7	3.9	3.5	3.7	3.2
	Mix- 13	Vanilla	4.6	2.5	2.3	2.6	2.6	2.6	2.8
	Mix- 13	STC	3.5	2.9	2.8	3.6	3.1	3.4	2.9
	Mix- 22	Vanilla	36.3	34.5	32	31.5	31.8	33.0	32.8
	Mix- 22	STC	33.0	34.2	31.7	33.2	31.6	33.7	32.8
	Mix- 100	Vanilla	4.3	2.6	2.3	2.6	2.7	2.8	2.7
	Mix- 100	STC	3.4	2.3	2.2	2.6	2.4	2.8	2.5
	Mix- 200	Vanilla	4.2	2.6	2.4	2.9	2.8	2.9	2.9
	Mix- 200	STC	3.6	2.3	2.1	2.6	2.7	2.6	2.6
	Mix- 577	Vanilla	4.8	2.8	2.4	2.8	2.9	2.8	2.9
	Mix- 577	STC	3.7	2.2	2.1	2.6	2.6	2.7	2.5
	Tag	Vanilla	6.2	-	-	-	-	-	-

Table A.4: Recognition results from Finnish EMIME tests. Error is measured in LER.

Appendix B

Statistical significance test results

The statistical significance test is done on a pair of recognition result transcriptions. The tables on the following pages show the most important tests. Each cell in the table shows a test between the system shown in the first column and a system shown in the first row of the table. The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed

The test used was the matched pair test as described in Section 4.2.2.

The tests are run using the *sc_stats* program.

SP eval set 1st pass	sep ML	mix0 STC	mix0 ML	mix3 STC	mix3 ML	mix6 STC	mix6 ML	mix9 STC	mix9 ML	mix13 STC	mix13 ML	mix100 STC	mix100 ML	mix200 STC	mix200 ML	mix577 STC	mix577 ML
sep STC	⇐	*	⇐	*	⇐	*	⇐	*	⇐	*	⇐	*	⇐	⇐	⇐	⇐	⇐
sep ML		⇑	*	⇑	*	⇑	*	⇑	*	⇑	*	⇑	*	⇑	*	⇑	*
mix0 STC			⇐	*	⇐	*	⇐	*	⇐	*	⇐	*	⇐	*	⇐	*	⇐
mix0 ML				⇑	*	⇑	⇐	⇑	*	⇑	*	⇑	*	⇑	⇐	⇑	*
mix3 STC					⇐	*	⇐	*	⇐	*	⇐	*	⇐	*	⇐	*	⇐
mix3 ML						⇑	*	⇑	*	⇑	*	⇑	*	⇑	*	⇑	*
mix6 STC							⇐	*	⇐	*	⇐	*	⇐	*	⇐	*	⇐
mix6 ML								⇑	*	⇑	⇑	⇑	*	⇑	*	⇑	*
mix9 STC									⇐	*	⇐	*	⇐	*	⇐	*	⇐
mix9 ML										⇑	*	⇑	*	⇑	*	⇑	*
mix13 STC											⇐	*	⇐	*	⇐	*	⇐
mix13 ML												⇑	*	⇑	⇐	⇑	*
mix100 STC													⇐	⇐	⇐	⇐	⇐
mix100 ML														⇑	*	⇑	*
mix200 STC															⇐	*	⇐
mix200 ML																⇑	*
mix577 STC																	⇐
mix577 ML																	

Table B.1: Statistically significant differences of recognition results with all viable system without adaptation on Speecon evaluation dataset. The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

SP eval set, CMLLR adaptation	sep ML CMLLR	mix0 STC CMLLR	mix0 ML CMLLR	mix3 STC CMLLR	mix3 ML CMLLR	mix6 STC CMLLR	mix6 ML CMLLR	mix9 STC CMLLR	mix9 ML CMLLR	mix13 STC CMLLR	mix13 ML CMLLR	mix100 STC CMLLR	mix100 ML CMLLR	mix200 STC CMLLR	mix200 ML CMLLR	mix577 STC CMLLR	mix577 ML CMLLR
sep STC CMLLR	*	⇐	*	*	*	*	*	⇐	*	*	*	⇑	*	⇑	*	⇑	*
sep ML CMLLR		⇐	*	⇐	⇐	⇐	⇐	⇐	*	⇐	*	⇑	*	⇑	*	*	⇐
mix0 STC CMLLR			⇑	*	*	*	*	*	*	*	*	⇑	⇑	⇑	*	⇑	*
mix0 ML CMLLR				*	*	*	*	⇐	*	*	*	⇑	*	⇑	*	⇑	*
mix3 STC CMLLR					*	*	*	*	*	*	*	⇑	*	⇑	*	⇑	*
mix3 ML CMLLR						*	*	*	*	*	*	⇑	*	⇑	*	⇑	*
mix6 STC CMLLR							*	*	*	*	*	⇑	*	⇑	*	⇑	*
mix6 ML CMLLR								*	*	*	*	⇑	*	⇑	*	⇑	*
mix9 STC CMLLR									*	*	*	⇑	⇑	⇑	⇑	⇑	*
mix9 ML CMLLR										*	*	⇑	*	⇑	*	⇑	*
mix13 STC CMLLR											*	⇑	*	⇑	*	⇑	*
mix13 ML CMLLR												⇑	*	⇑	*	⇑	*
mix100 STC CMLLR													⇐	*	⇐	*	⇐
mix100 ML CMLLR														⇑	*	⇑	*
mix200 STC CMLLR															⇐	*	⇐
mix200 ML CMLLR																⇑	*
mix577 STC CMLLR																	⇐
mix577 ML CMLLR																	

Table B.2: Statistically significant differences of recognition results with all viable systems and intra-lingual CMLLR adaptation on Speecon evaluation dataset.
The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

WSJ eval 1st pass	sep ML	mix0 STC	mix0 ML	mix3 STC	mix3 ML	mix6 STC	mix6 ML	mix9 STC	mix9 ML	mix13 STC	mix13 ML	mix100 STC	mix100 ML	mix200 STC	mix200 ML	mix577 STC	mix577 ML
sep STC	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐
sep ML		⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	*	⇐	*	⇐	*	⇐
mix0 STC			*	*	⇐	*	*	*	*	*	*	⇑	*	⇑	*	⇑	*
mix0 ML				⇑	*	⇑	*	*	*	*	*	⇑	⇑	⇑	⇑	⇑	⇑
mix3 STC					⇐	*	*	*	*	*	*	⇑	*	⇑	*	⇑	*
mix3 ML						⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix6 STC							*	*	*	*	*	⇑	*	⇑	*	⇑	*
mix6 ML								*	*	*	*	⇑	⇑	⇑	⇑	⇑	⇑
mix9 STC									*	*	*	⇑	⇑	⇑	⇑	⇑	⇑
mix9 ML										*	*	⇑	⇑	⇑	⇑	⇑	⇑
mix13 STC											*	⇑	⇑	⇑	⇑	⇑	*
mix13 ML												⇑	⇑	⇑	⇑	⇑	⇑
mix100 STC													*	*	*	*	*
mix100 ML														*	*	*	*
mix200 STC															*	*	*
mix200 ML																*	*
mix577 STC																	*
mix577 ML																	

Table B.3: Statistically significant differences of recognition results with all viable systems without adaptation on the WSJ evaluation set.
The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

WSJ eval CMLLR adapted	sep ML	mix0 STC	mix0 ML	mix3 STC	mix3 ML	mix6 STC	mix6 ML	mix9 STC	mix9 ML	mix13 STC	mix13 ML	mix100 STC	mix100 ML	mix200 STC	mix200 ML	mix577 STC	mix577 ML
sep STC	*	*	*	*	*	*	*	*	*	*	*	⇑	*	⇑	*	⇑	*
sep ML		⇐	⇐	⇐	⇐	*	*	⇐	⇐	⇐	⇐	*	*	*	*	*	⇐
mix0 STC			*	*	*	*	*	*	*	*	*	⇑	*	⇑	⇑	⇑	*
mix0 ML				*	*	*	*	*	*	*	*	⇑	*	⇑	*	⇑	*
mix3 STC					*	*	*	*	*	*	*	⇑	*	⇑	⇑	⇑	*
mix3 ML						*	*	*	*	*	*	⇑	*	⇑	*	⇑	*
mix6 STC							*	*	*	*	*	⇑	*	⇑	*	⇑	*
mix6 ML								*	*	*	*	⇑	*	⇑	*	*	*
mix9 STC									*	*	*	⇑	⇑	⇑	⇑	⇑	*
mix9 ML										*	*	⇑	*	⇑	*	⇑	*
mix13 STC											*	⇑	*	⇑	⇑	⇑	*
mix13 ML												⇑	*	⇑	⇑	⇑	*
mix100 STC													⇐	*	⇐	*	⇐
mix100 ML														⇑	*	*	*
mix200 STC															⇐	*	⇐
mix200 ML																*	*
mix577 STC																	*
mix577 ML																	

Table B.4: Statistically significant differences of recognition results with all viable systems with intra-lingual CMLLR adaptation on the WSJ evaluation dataset. The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

EMIME English 1st pass	sep ML	mix0 STC	mix0 ML	mix3 STC	mix3 ML	mix6 STC	mix6 ML	mix9 STC	mix9 ML	mix13 STC	mix13 ML	mix100 STC	mix100 ML	mix200 STC	mix200 ML	mix577 STC	mix577 ML
sep STC	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	*	⇐	*	⇐	*	⇐
sep ML		*	⇐	*	⇐	*	⇐	*	⇐	*	⇐	⇑	*	⇑	*	⇑	*
mix0 STC			⇐	*	⇐	*	⇐	*	⇐	*	⇐	⇑	*	⇑	*	⇑	*
mix0 ML				⇑	*	⇑	⇐	⇑	*	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑
mix3 STC					⇐	*	⇐	*	⇐	*	⇐	⇑	*	⇑	*	⇑	*
mix3 ML						⇑	*	⇑	*	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑
mix6 STC							⇐	*	⇐	*	⇐	⇑	*	⇑	*	⇑	*
mix6 ML								⇑	*	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑
mix9 STC									⇐	*	⇐	⇑	*	⇑	*	⇑	*
mix9 ML										⇑	*	⇑	⇑	⇑	⇑	⇑	⇑
mix13 STC											⇐	⇑	*	*	*	⇑	*
mix13 ML												⇑	⇑	⇑	⇑	⇑	⇑
mix100 STC													⇐	*	⇐	*	⇐
mix100 ML														⇑	*	⇑	*
mix200 STC															⇐	*	⇐
mix200 ML																⇑	*
mix577 STC																	⇐
mix577 ML																	

Table B.5: Statistically significant differences in recognition performance with all viable systems without adaptation and EMIME English dataset.
The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

EMIME English with CMLLR adaptation	sep ML	mix0 STC	mix0 ML	mix3 STC	mix3 ML	mix6 STC	mix6 ML	mix9 STC	mix9 ML	mix13 STC	mix13 ML	mix100 STC	mix100 ML	mix200 STC	mix200 ML	mix577 STC	mix577 ML
sep STC	*	⇐	⇐	⇐	*	⇐	⇐	⇐	⇐	⇐	⇐	⇑	*	*	*	⇑	*
sep ML		⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	*	*	*	*	*	*
mix0 STC			*	*	⇑	*	⇑	*	*	*	⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix0 ML				*	⇑	*	*	*	*	*	*	⇑	⇑	⇑	⇑	⇑	⇑
mix3 STC					⇑	*	⇑	*	*	*	⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix3 ML						⇐	*	⇐	*	⇐	*	⇑	⇑	⇑	⇑	⇑	⇑
mix6 STC							⇑	*	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix6 ML								*	*	*	*	⇑	⇑	⇑	⇑	⇑	⇑
mix9 STC									*	*	⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix9 ML										*	*	⇑	⇑	⇑	⇑	⇑	⇑
mix13 STC											*	⇑	⇑	⇑	⇑	⇑	⇑
mix13 ML												⇑	⇑	⇑	⇑	⇑	⇑
mix100 STC													*	*	*	*	*
mix100 ML														*	*	*	*
mix200 STC															*	*	*
mix200 ML																*	*
mix577 STC																	*
mix577 ML																	

Table B.6: Statistically significant differences of recognition results with all viable systems and intra-lingual adapation on the English EMIME dataset.
The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

EMIME Finnish 1st pass	sep ML	mix0 STC	mix0 ML	mix3 STC	mix3 ML	mix6 STC	mix6 ML	mix9 STC	mix9 ML	mix13 STC	mix13 ML	mix100 STC	mix100 ML	mix200 STC	mix200 ML	mix577 STC	mix577 ML
sep STC	⇐	⇐	⇐	⇐	⇐	⇐	⇐	*	⇐	*	⇐	*	⇐	*	⇐	*	⇐
sep ML		*	⇐	*	⇐	*	⇐	⇑	⇐	⇑	*	⇑	*	⇑	*	⇑	⇐
mix0 STC			⇐	*	⇐	*	⇐	*	⇐	⇑	⇐	⇑	*	⇑	*	*	⇐
mix0 ML				⇑	*	⇑	*	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix3 STC					⇐	*	⇐	*	⇐	⇑	⇐	⇑	⇐	*	*	*	⇐
mix3 ML						⇑	*	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑	⇑	*
mix6 STC							⇐	⇑	⇐	⇑	⇐	⇑	*	⇑	*	⇑	⇐
mix6 ML								⇑	*	⇑	⇑	⇑	⇑	⇑	⇑	⇑	*
mix9 STC									⇐	⇑	⇐	*	⇐	*	⇐	*	⇐
mix9 ML										⇑	*	⇑	⇑	⇑	⇑	⇑	*
mix13 STC											⇐	*	⇐	*	⇐	*	⇐
mix13 ML												⇑	*	⇑	⇑	⇑	*
mix100 STC													⇐	*	⇐	*	⇐
mix100 ML														⇑	*	⇑	*
mix200 STC															⇐	*	⇐
mix200 ML																⇑	⇐
mix577 STC																	⇐
mix577 ML																	

Table B.7: Statistically significant differences in recognition results with all viable systems and no adaptation on the Finnish EMIME dataset.
The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

EMIME Finnish CMLLR	sep ML	mix0 STC	mix0 ML	mix3 STC	mix3 ML	mix6 STC	mix6 ML	mix9 STC	mix9 ML	mix13 STC	mix13 ML	mix100 STC	mix100 ML	mix200 STC	mix200 ML	mix577 STC	mix577 ML
sep STC	⇑	*	⇑	*	⇑	*	⇑	*	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑	*
sep ML		*	*	*	*	*	*	*	*	⇐	*	⇑	*	⇑	*	⇑	*
mix0 STC			⇑	*	⇑	*	⇑	*	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑	*
mix0 ML				*	*	⇐	*	⇐	*	⇐	*	*	*	*	*	*	*
mix3 STC					⇑	*	⇑	*	*	*	*	⇑	*	⇑	*	⇑	*
mix3 ML						⇐	*	⇐	*	⇐	*	*	*	*	*	*	*
mix6 STC							⇑	*	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑	*
mix6 ML								⇐	*	⇐	*	*	*	*	*	*	*
mix9 STC									⇑	*	⇑	⇑	⇑	⇑	⇑	⇑	*
mix9 ML										⇐	*	*	*	*	*	*	*
mix13 STC											⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix13 ML												*	*	*	*	*	*
mix100 STC													*	*	⇐	*	⇐
mix100 ML														*	*	*	*
mix200 STC															⇐	*	⇐
mix200 ML																⇑	*
mix577 STC																	⇐
mix577 ML																	

Table B.8: Statistically significant differences in recognition results with all viable systems with intra-lingual adaptation on Finnish EMIME dataset.
The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

EMIME English, ML-sep and ML-mix100 cross lingual adaptation	sep ML	sep STC phoneloop rt.	sep STC phoneloop 1	sep ML phoneloop rt	sep ML phoneloop 1	mix100 STC	mix100 ML	mix100 STC phoneloop rt	mix100 STC phoneloop 1	mix100 ML phoneloop rt	mix100 ML phoneloop 1	mix100 STC cross-adapted rt	mix100 STC cross-adapted 1	mix100 ML cross-adapted rt	mix100 ML cross-adapted 1
sep STC	⇐	*	*	*	*	*	⇐	*	*	*	*	*	*	*	*
sep ML		⇑	⇑	⇑	⇑	⇑	*	⇑	⇑	⇑	⇑	*	⇑	⇑	⇑
sep STC pl. rt			*	*	*	*	⇐	*	*	*	*	*	*	*	*
sep STC pl. 1				*	*	⇐	⇐	*	*	*	*	⇐	⇐	*	⇐
sep ML pl. rt					*	⇐	⇐	*	*	*	*	⇐	*	*	⇐
sep ML pl. 1						*	⇐	*	*	⇑	*	*	*	*	*
mix100 STC							⇐	⇑	⇑	⇑	*	*	*	*	*
mix100 ML								⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix100 STC pl. rt									*	*	*	⇐	⇐	⇐	⇐
mix100 STC pl. 1										*	*	⇐	⇐	*	⇐
mix100 ML pl. rt											⇐	⇐	*	⇐	⇐
mix100 ML pl.												*	*	*	*
mix100 STC cr. rt													*	*	*
mix100 STC cr. 1														*	*
mix100 ML cr. rt															*
mix100 ML cr. 1															

Table B.9: Statistically significant differences in phoneloop (pl.) and cross-lingual (cr.), full regression tree (rt) or single (1) adaptation with ML-sep and ML-mix$_{100}$ systems on the English EMIME dataset.
The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

EMIME Finnish, ML-sep and ML-mix100 cross lingual adaptation	sep ML	sep STC phoneloop rt	sep STC phoneloop 1	sep ML phoneloop rt	sep ML phoneloop 1	mix100 STC	mix100 ML	mix100 STC phoneloop rt	mix100 STC phoneloop 1	mix100 ML phoneloop rt	mix100 ML phoneloop 1	mix100 STC cross rt	mix100 STC cross 1	mix100 ML cross rt	mix100 ML cross 1
sep STC	⇐	*	⇑	⇑	⇑	*	⇐	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
sep ML		⇑	⇑	⇑	⇑	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
sep STC pl. rt			⇑	⇑	⇑	*	⇐	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
sep STC pl. 1				*	*	*	⇐	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
sep ML pl. rt					*	⇐	⇐	*	⇑	*	*	*	⇑	*	*
sep ML pl. 1						⇐	⇐	*	⇑	*	*	⇑	⇑	*	*
mix100 STC							⇐	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix100 ML								⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix100 STC pl. rt									⇑	*	*	*	⇑	*	*
mix100 STC pl. 1										⇐	*	*	*	*	*
mix100 ML pl. rt											*	*	⇑	*	*
mix100 ML pl. 1												*	⇑	*	*
mix100 STC cr. rt													*	*	*
mix100 STC cr. 1														*	*
mix100 ML cr. rt															*
mix100 ML cr. 1															

Table B.10: Statistically significant differences in phoneloop (pl.) and cross-lingual (cr.), full regression tree (rt) or single (1) adaptation with ML-sep and ML-mix$_{100}$ systems on the Finnish EMIME dataset.
The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

EMIME English, ML-sep and ML-mix13 cross lingual adaptation	sep ML	sep STC phoneloop rt	sep STC phoneloop 1	sep ML phoneloop rt	sep ML phoneloop 1	mix13 STC	mix13 ML	mix13 STC phoneloop rt	mix13 STC phoneloop 1	mix13 ML phoneloop rt	mix13 ML phoneloop 1	mix13 STC cross rt	mix13 STC cross 1	mix13 ML cross rt	mix13 ML cross 1
sep STC	⇐	*	*	*	*	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐
sep ML		⇑	⇑	⇑	⇑	*	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐
sep STC pl. rt			*	*	*	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐
sep STC pl. 1				*	*	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐
sep ML pl. rt					*	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐
sep ML pl. 1						⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐
mix13 STC							⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐
mix13 ML								⇐	⇐	*	*	⇐	⇐	⇐	⇐
mix13 STC pl. rt									⇑	⇑	⇑	⇐	*	⇑	⇑
mix13 STC pl. 1										⇑	⇑	⇐	⇐	⇑	*
mix13 ML pl. rt											*	⇐	⇐	⇐	⇐
mix13 ML pl. 1												⇐	⇐	⇐	⇐
mix13 STC cr. rt													⇑	⇑	⇑
mix13 STC cr. 1														⇑	⇑
mix13 ML cr. rt															⇐
mix13 ML cr. 1															

Table B.11: Statistically significant differences in phoneloop (pl.) and cross-lingual (cr.), full regression tree (rt) or single (1) adaptation with ML-sep and ML-mix$_{13}$ systems on the English EMIME dataset.
The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

EMIME Finnish, ML-sep and ML-mix13 cross lingual adaptation	sep ML	sep STC phoneloop rt	sep STC phoneloop 1	sep ML phoneloop rt	sep ML phoneloop 1	mix13 STC	mix13 ML	mix13 STC phoneloop rt	mix13 STC phoneloop 1	mix13 ML phoneloop rt	mix13 ML phoneloop 1	mix13 STC cross rt	mix13 STC cross 1	mix13 ML cross rt	mix13 ML cross 1
sep STC	⇐	*	⇑	⇑	⇑	*	⇐	*	⇑	⇑	⇑	*	⇑	⇑	⇑
sep ML		⇑	⇑	⇑	⇑	⇑	*	⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
sep STC pl. rt			⇑	⇑	⇑	*	⇐	*	⇑	⇑	⇑	*	*	⇑	⇑
sep STC pl. 1				*	*	*	⇐	*	*	⇑	*	⇐	*	⇑	⇑
sep ML pl. rt					*	⇐	⇐	⇐	*	*	*	⇐	*	*	*
sep ML pl. 1						⇐	⇐	⇐	*	⇑	*	⇐	*	*	*
mix13 STC							⇐	*	⇑	⇑	⇑	*	⇑	⇑	⇑
mix13 ML								⇑	⇑	⇑	⇑	⇑	⇑	⇑	⇑
mix13 STC pl. rt									⇑	⇑	⇑	*	⇑	⇑	⇑
mix13 STC pl. 1										⇑	*	⇐	*	*	*
mix13 ML pl. rt											⇐	⇐	⇐	*	*
mix13 ML pl. 1												⇐	*	*	*
mix13 STC cr. rt													⇑	⇑	⇑
mix13 STC cr. 1														⇑	⇑
mix13 ML cr. rt															*
mix13 ML cr. 1															

Table B.12:

Statistically significant differences in phoneloop (pl.) and cross-lingual (cr.), full regression tree (rt) or single (1) adaptation with ML-sep and ML-mix$_{13}$ systems on the Finnish EMIME dataset.
The arrow shows which of the two compared systems performs better. A star denotes that no significant performance differences can be claimed. The best systems have been shaded - These systems are as good or better than any others.

REFERENCES

Andersen Ove, Dalsgaard Paul, & Barry William. 1993. Data-driven identification of poly- and mono-phonemes for four european languages. *EUROSPEECH-1993*, 759–762.

Aubert X., Dugast C., Ney H., & Steinbiss V. 1994. Large Vocabulary Continuous Speech Recognition of Wall Street Journal Data. *ICASSP'94*, 129–132.

Chittka Lars, & Brockmann Axel. 2005. Perception Space—The Final Frontier. *PLoS Biol*, **3**(4), e137.

Creutz Mathias. 2006. *Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition.* Ph.D. thesis, Helsinki University of Technology.

Denes Peter B., , & Pinson Elliot N. 1970. *The Speech Chain The Physics and Biology of Spoken Language.* New York: Bell Telephone Laboratories for Educational Use.

Deshmukh O., Espy-Wilson C.Y., & Juneja A. 2002. Acoustic-phonetic speech parameters for speaker-independent speech recognition. *Pages I–593–I–596 vol.1 of: Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, vol. 1.

Dines John, Yamagishi Junichi, & King Simon. 2009. Measuring the gap between HMM-based ASR and TTS. *INTERSPEECH-2009.*

Gales M.J.F. 1996. *The generation and use of regression class trees for MLLR adaptation, Technical Report.*

Gales M.J.F. 1998. Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition. *Computer Speech and Language*, **12**, 75–98.

Gales M.J.F. 1999. Semi-Tied Covariance Matrices For Hidden Markov Models. *IEEE Transactions on Speech and Audio Processing*, **7**, 272–281.

Gillick L., & Cox S.J. 1989 (May). Some statistical issues in the comparison of speech recognition algorithms. *Pages 532–535 vol.1 of: Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on.*

Gokcen S., & Gokcen J.M. 1997. A multilingual phoneme and model set: toward a universal base for automatic speech recognition. *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, Dec, 599–605.

Goldstein E. Bruce. 2001. *Sensation and Perception.* 6th edn. Wadworth Publishing Company.

Hershey J.R., & Olsen P.A. 2007 (April). Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models. *Pages IV–317–IV–320 of: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4.

Hirsch Hans-Günter, & Pearce David. 2000. The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions. *ASR-2000*, 181–188.

Hirsimäki T. Creutz M. Siivola V. Kurimo M. Virpioja S. & Pylkkönen. J.̇2006. Unlimited vocabulary speech recognition with morph language models applied to Finnish. *Computer Speech & Language*, **20**(4), 515–541.

Hirsimäki Teemu. 2009. *Advances in unlimited-vocabulary speech recognition for morphologically rich languages.* Ph.D. thesis, Helsinki University of Technology.

Imperl Bojan, & Horvat Bogomir. 1999. The clustering algorithm for the definition of multilingual set of context dependent speech models. *EUROSPEECH-1999*, 887–890.

Imperl Bojan, Kacic Zdravko, Horvat Bogomir, & Zgank Andrej. 2003. Clustering of triphones using phoneme similarity estimation for the definition of a multilingual set of triphones. *Speech Communication*, **39**(3-4), 353 – 366.

Iskra D., Grosskopf B., Marasek K., Heuvel H.V.D., Diehl F., & Kiessling A. 2002. Speecon - speech databases for consumer devices: Database specification and validation. *Pages 329–333 of: in Proc. LREC, 2002.*

Jalanko Matti. 1980. *Studies of Learning Projective Methods in Automatic Speech Recognition.* Ph.D. thesis, Helsinki University of Technology.

Kallasjoki Heikki. 2009. *Methods for Spectral Envelope Estimation in Noise Robust Speech Recognition.* M.Sc.Tech. thesis, Helsinki University of Technology.

Keronen Sami. 2009. *Parallel Model Combination in large vocabulary continuous speech recognition.* M.Sc.Tech. thesis, Helsinki University of Technology.

Kinjo T., & Funaki K. 2006 (Nov.). On HMM Speech Recognition Based on Complex Speech Analysis. *Pages 3477–3480 of: IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on.*

Kullback S., & Leibler R. A. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, **22**, 79–86.

Kumar C. Santhosh, Mohandas V. P., & Li Haizhou. 2005. Multilingual speech recognition: a unified approach. *INTERSPEECH-2005*, 3357–3360.

Levenshtein V. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics*, 707–710.

Lyu Dau-cheng, & Lyu Ren-yuan. 2008 (31 2008-April 4). Optimizing the acoustic modeling from an unbalanced bi-lingual corpus. *Pages 4301–4304 of: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on.*

Mandal Arindam, Ostendorf Mari, & Stolcke Andreas. 2006. Speaker clustered regression-class trees for MLLR adaptation. *INTERSPEECH-2006*, 1133–1136.

Mansikkaniemi André. 2010. *Acoustic Model and Language Model Adaptation for a Mobile Dictation Service.* M.Sc.Tech. thesis, Helsinki University of Technology.

Mill John Stuart. 1861. Utilitarianism. *Fraser's Magazine.*

Paul Douglas B., & Baker Janet M. 1992. The design for the wall street journal-based CSR corpus. *Pages 357–362 of: HLT '91: Proceedings of the workshop on Speech and Natural Language.* Morristown, NJ, USA: Association for Computational Linguistics.

Pols Louis C.W., Wang Xue, & ten Bosch Louis F.M. 1996. Modelling of phone duration (using the TIMIT database) and its potential benefit for ASR. *Speech Communication*, **19**(2), 161 – 176.

Pye D., & Woodland P.C. 1997 (Apr). Experiments in speaker normalisation and adaptation for large vocabulary speech recognition. *Pages 1047–1050 vol.2 of: Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2.

Pylkkönen Janne. 2005. An Efficient One-pass Decoder for Finnish Large Vocabulary Continuous Speech Recognition. *2nd Baltic Conference on Human Language Technologies (HLT'2005), Tallinn, Estonia*, Apr, 167–172.

Pylkkönen Janne. 2009 (September). Investigations on Discriminative Training in Large Scale Acoustic Model Estimation. *Pages 220–223 of: Proc. Interspeech.*

Rabiner Lawrence. 1989. A tutorial on HMM and selected applications in speech recognition. *Proceedings of the IEEE*, **77**(2), 257–286.

Remes Ulpu. 2007. *Speaker-Based Segmentation and Adaptation in Automatic Speech Recognition.* M.Sc.Tech. thesis, Helsinki University of Technology.

Schultz Tanja, & Kirchhoff Katrin. 2006. *Multilingual Speech Processing.* Elsevier.

Schultz Tanja, & Waibel Alex. 1998. Multilingual and crosslingual speech recognition. *Pages 259–262 of: Proceedings of the DARPA Broadcast News Transcr-Proceedings of the DARPA Broadcast News Transcription and Understanding, DARPA 1998, Lansdowne Virginia.*

Shannon C. E. 2001. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, **5**(1), 3–55.

Siivola Vesa. 2007. *Language Models for Automatic Speech Recognition: Construction and Complexity Control.* Ph.D. thesis, Helsinki University of Technology.

Sooful Jayren J., & Botha Elizabeth C. 2002. Comparison of acoustic distance measures for automatic cross-language phoneme mapping. *ICSLP-2002*, 521–524.

Varjokallio Matti. 2007. *Subspace Methods gor Gaussian Mixture Models in Automatic Speech Recognition.* M.Sc.Tech. thesis, Helsinki University of Technology.

Vihola Matti. 2001. *Dissimilarity Measures for Hidden Markov Models and Their Application in Multilingual Speech Recognition.* Master's Thesis, Tampere University of Technology.

Weng Fuliang, Bratt Harry, Neumeyer Leonardo, & Stolcke Andreas. 1997. A study of multilingual speech recognition. *EUROSPEECH-1997*, 359–362.

Yamagishi Junichi, Usabaev Bela, King Simon, Watts Oliver, Dines John, Tian Jilei, Hu Rile, Oura Keiichiro, Tokdua Keiichi, Karhila Reima, & Kurimo Mikko. 2009. Thousands of Voices for HMM-based Speech Synthesis. *In: Proc. Interspeech.*

Yan Qin, & Vaseghi S. 2002. A comparative analysis of UK and US English accents in recognition and synthesis. *Pages I–413–I–416 vol.1 of: Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, vol. 1.

Young S. J., Evermann G., Gales M. J. F., Hain T., Kershaw D., Moore G., Odell J., Ollason D., Povey D., Valtchev V., & Woodland P. C. 2006. *The HTK Book, version 3.4.* Cambridge, UK: Cambridge University Engineering Department.

Zigelboim G., & Shallom I.D. 2006 (Oct.). A comparison Study of Cepstral Analysis with Applications to Speech Recognition. *Pages 30–33 of: Information Technology: Research and Education, 2006. ITRE '06. International Conference on.*