Helsinki University of Technology
Department of Engineering Physics and Mathematics

Ulpu Remes

Speaker-Based Segmentation and Adaptation in Automatic Speech Recognition

Master's thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology

Espoo, April 26, 2007

Supervisor: professor Erkki Oja
Instructor: docent Mikko Kurimo

Preface

This work was done in the Laboratory of Computer and Information Science at the Helsinki University of Technology during the years 2006 and 2007. It was supported by the Academy of Finland and the Finnish National Technology Agency (Tekes) in the projects *New adaptive and learning methods in speech recognition* and *New methods and applications for speech technology*. I thank professor Erkki Oja for supervising the thesis. I thank my instructor docent Mikko Kurimo for the opportunity to work in the speech group and for the valuable advice he has given. This work would not have been possible without the prior work done in the speech group, and thus, I have the current and former speech group members to thank. I would like to take this opportunity to especially thank Janne Pylkkönen, Teemu Hirsimäki and Vesa Siivola who have helped me with all those various problems that I have encountered during my time in the laboratory. Also, Kalle Palomäki is to thank for the time he has taken to read this thesis and for his comments that helped to improve the work. I thank Tommi, and I thank my friends who shared a cup of coffee with me when I needed their kind words for encouragement.

Ulpu Remes
Espoo, April 26, 2007

HELSINKI UNIVERSITY OF TECHNOLOGY ABSTRACT OF MASTER'S THESIS

Author:	Ulpu Remes
Department:	Engineering Physics and Mathematics
Major subject:	Computer and Information Science
Minor subject:	Automation and Control Technology
Title:	Speaker-Based Segmentation and Adaptation in Automatic Speech Recognition
Title in Finnish:	Puhujakohtainen segmentointi ja mukautuminen automaattisessa puheentunnistuksessa
Chair:	T-61 Computer and Information Science
Supervisor:	Professor Erkki Oja
Instructor:	Docent Mikko Kurimo

Abstract:

With proper training, automatic speech recognition works quite well when tested in conditions similar to the training conditions, but with a new speaker or a new environment the system performance often degrades. Speaker-based adaptation alters the speech recognition system to better match a specific speaker and thus improves the speech recognition results. In order to use speaker adaptation, the speech recognition system would, however, need to know who spoke and when. When no prior information about the speakers is available, automatic speaker-based segmentation methods are needed. Speaker segmentation should divide the speech data to speaker turns, that indicate when speakers change, and then label the speaker turns according to the speaker.

In this work, a metric-based approach is presented for speaker change detection and two methods for clustering and labelling the detected speaker turns are discussed. The methods are evaluated in speaker segmentation and adaptation in large vocabulary continuous speech recognition task with Finnish and English broadcast news audio. The results show that the coupled automatic speaker segmentation and adaptation developed in this thesis improves the overall speech recognition performance significantly.

Pages: 68 Keywords: speaker adaptation, speaker segmentation

Department fills
Approved: Library code:

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Ulpu Remes
Osasto:	Teknillisen fysiikan ja matematiikan osasto
Pääaine:	Informaatiotekniikka
Sivuaine:	Automaatio- ja systeemitekniikka
Työn nimi:	Puhujakohtainen segmentointi ja mukautuminen automaattisessa puheentunnistuksessa
Title in English:	Speaker-Based Segmentation and Adaptation in Automatic Speech Recognition
Professuuri:	T-61 Informaatiotekniikka
Työn valvoja:	Professori Erkki Oja
Työn ohjaaja:	Dosentti Mikko Kurimo

Tiivistelmä:

Huolella opetettu puheentunnistusjärjestelmä toimii varsin hyvin opetusvaiheesta tutuissa olosuhteissa, mutta uusi puhuja tai ympäristö usein heikentää järjestelmän suorituskykyä. Puheentunnistustuloksia voidaan siksi parantaa antamalla järjestelmän mukautua kullekin puhujalle paremmin soveltuvaksi. Puhujakohtainen mukautuminen ei kuitenkaan ole mahdollista, ellei järjestelmä tiedä, kuka milloinkin on äänessä. Puhetallenteita ei ole tavallisesti jaettu osiin puhujan mukaan, joten tarvitaan automaattisia menetelmiä puhujien erotteluun. Puhujakohtaisen segmentoinnin tavoitteena on jakaa annettu aineisto puheenvuoroihin, joista voidaan sekä lukea puhujanvaihdosten ajankohdat että seurata kulloinkin vuorossa olevaa puhujaa.

Tässä työssä selvitetään, kuinka puhujanvaihdokset voidaan havaita erilaisuusmittoihin perustuvilla menetelmillä, sekä tarkastellaan kahta nimeämättömien puheenvuorojen ryhmittelyyn soveltuvaa menetelmää. Ryhmittely on keino koota ja nimetä havaitut puheenvuorot puhujan mukaan, kun puhujia tai heidän määräänsä ei tunneta ennakkoon. Puhujasegmentoinnin toimivuutta kokeillaan yhdessä puhujakohtaisen mukautumisen kanssa laajan sanaston jatkuvan puheen tunnistuksessa. Koeaineistona käytetään otteita suomen- ja englanninkielisistä uutislähetyksistä. Saadut tulokset osoittavat, että puhujakohtainen segmentointi ja mukautuminen yhdessä parantavat puheentunnistustuloksia merkittävästi.

Sivumäärä: 68	Avainsanat: puhujakoht. mukautuminen, puhujasegmentointi

Täytetään osastolla

Hyväksytty:	Kirjasto:

Contents

Symbols and abbreviations

$\mathbf{m}$	Gaussian sample mean
$\mathbf{o}(\tau)$	Observed features at time τ
$\mathbf{S}$	Gaussian sample covariance matrix
T	Speaker change detection threshold
$\mathbf{W}$	Linear transformation matrix
λ	Scaling parameter $\lambda \in [0, 1]$
Λ	Model parameters
$\boldsymbol{\mu}$	Gaussian mean
$\boldsymbol{\Sigma}$	Gaussian covariance matrix
ACP	Average cluster purity
ASP	Average speaker purity
BIC	Bayesian information criterion
EM	Expectation-maximisation
FAR	False acceptance rate
FFT	Fast Fourier transform
FRR	False rejection rate
GMM	Gaussian mixture model
CMLLR	Constrained maximum likelihood linear regression
GLR	Generalised likelihood ratio
HMM	Hidden Markov model
KL	Kullback-Leibler
LER	Letter error rate
LVCSR	Large vocabulary continuous speech recognition
MAP	Maximum a posteriori
MFCC	Mel-frequency cepstral coefficients
ML	Maximum likelihood
MLLR	Maximum likelihood linear regression
SCD	Speaker change detection
VOA	Voice of America: English broadcast news
WER	Word error rate
YLE	Finnish Broadcasting company: Finnish television news

Chapter 1

Introduction

Speech recognition has long been a dream that is just about to come true. There are, after all, applications where speech recognition works quite well, but most often these have the system trained to recognise certain commands rather than continuous speech with unlimited vocabulary, and to have a computer convert generic language into written text still remains a formidable challenge. Speech recognition with unlimited vocabulary is difficult, because without any limitations, there is no context either: for a machine, it seems perfectly sensible to have a cup of coffee with dream and sugar. Different speakers and environments increase the chance to misunderstand and hence make the speech recognition task even more challenging.

Speech recognition actually works quite well when trained for one speaker in low-noise conditions, but change the conditions, and the system performance will surely degrade. Thus, robustness is one major question in speech recognition, and a question with many answers it is. We may train the system with speech data collected from various environments in order to improve the noise robustness, and similarly, we can use data from several different speakers to make the system speaker-independent. However, speaker independent systems are not quite as accurate as speaker dependent systems, for they model the average speaker, and although some speakers come very close to this model, equally many are not well-represented [50].

We truly make automatic speech recognition difficult, for we all sound different even if the words uttered are the same. Sometimes we notice this ourselves, and sometimes not. We can observe cultural differences in intonation and pronunciation, for example, and we notice environmental differences like traffic noise or poor telephone connection. However, differences inherent to anatomical characteristics like vocal tract shape and length we do not pay any attention to, and yet for a speech recogniser, they are as fatal as any other differences.

Speaker adaptation techniques alter the speaker-independent model to better match a specific speaker. Speakers still need to provide the system with some speech data, but speaker adaptation does not need anywhere near as much data as training would

do, and if necessary, it is possible to do adaptation on-line. A common approach to speaker-based adaptation would be linear transformation methods like constrained maximum likelihood linear regression (CMLLR). In CMLLR, the features chosen to represent the speech signal are treated with speaker-specific linear transformations to have them better match the speaker-independent model [21]. CMLLR and other speaker adaptation methods have proven very efficient in improving speech recognition results.

In order to estimate the speaker-specific transformations, we would, however, need to know the speakers. We do not need to recognise the speakers in the sense that we would find a name or some other identity for each speaker, but we need to know when speakers change, and we need to know if some have spoken several times. In news broadcasts, for example, where speakers change frequently and new speakers appear, such information is not often available in advance, but speaker segmentation methods are needed before we can apply speaker adaptation.

Speaker-based segmentation is expected to divide the audio to speaker turns that determine which speaker is active at the time. Thus, each turn is associated with one speaker only, whereas speakers may take several turns. Given no prior information on the speakers, we must first find the speaker change boundaries that define where one turn ends and another begins, and then label the speaker turns correctly. Speaker-based segmentation is also essential in many speech technology applications like retrieval and browsing of large automatically transcribed audio files and the analysis of spoken dialogs and multi-party meetings.

In this work, we are primarily interested in finding a speaker segmentation method that could provide the speaker turns for speaker-based adaptation. To this end, we discuss some popular speaker change detection methods and distance metrics, and compare the two most common approaches in speaker turn labelling: we have a well-known speaker model based method [9] and a new method that couples speaker-based segmentation and adaptation. Our method is quite similar to that proposed in [76], where speaker-dependent models were utilised in labelling speaker turns. The main difference between that method and the method proposed here is that we use speaker-dependent feature transformations as opposed to speaker-dependent models.

Speaker-based segmentation and adaptation are tested in speech recognition framework with the large vocabulary continuous speech recognition (LVCSR) system developed in the Laboratory of Computer and Information Science at the Helsinki University of Technology. The speech recogniser can be trained for different languages and has been tested with Finnish and Estonian [41] for example. The system does speaker adaptation and normalisation if given speaker-labelled audio files. In this work, a metric-based speaker change detection method and the two methods discussed for speaker turn labelling were implemented in the system to enable automatic speaker-based segmentation. Several tests were conducted with Finnish and English broadcast news audio to measure the system performance. The results indicate that speaker-based segmentation and adaptation significantly improve the speech recogni-

tion results, and the new speaker labelling method based on speaker-specific feature transformations is found to have a performance comparable to that of the well-known speaker model based method [9].

This thesis begins with a short introduction to speech recognition in Chapter 2. Emphasis is on acoustic modelling, for the speaker adaptation techniques discussed in Chapter 3 operate on the acoustic models or features. Speaker segmentation methods are presented in Chapter 4, and in Chapter 5 we test speaker-based segmentation and adaptation with English broadcast news data and Finnish television news audio. We evaluate the results from speaker segmentation as well as speaker adaptation or speech recognition perspective. Chapter 6 then summarises the results and discusses possible future improvements, thus concluding this work.

Chapter 2

Speech Recognition

Large vocabulary continuous speech recognition (LVCSR) is among the most challenging areas in automatic speech recognition. Continuous speech is harder to recognise than isolated words, for there are no pauses to indicate when a word should end and another begin. And of course, the larger the vocabulary, the easier it is to confuse the recogniser. It is hard for humans to understand this, for we do not remember the time when we were still learning, when spoken words made no sense, had no ideas attached to them. Given that automatic systems most often do not understand natural language, it is quite surprising, actually, how well they manage speech recognition. Following discussions related to automatic speech recognition are largely based on [59].

Speech is most often represented with the signal waveform, but this representation is not well-suited for speech recognition as it is quite complex and passes on more information than the spoken words alone: for example, one can read emotions from the speech signal waveform, for emotions reflect to fundamental frequency, amplitude variations and speech rate [71]. We do not wish to recognise emotions, so we would like to represent the speech signal with features that conserve only the information that is essential for speech recognition. LVCSR systems then seek to find the word sequence W that has most likely produced the observed features $\mathbf{O} = \{\mathbf{o}(\tau)\}$, where $\mathbf{o}(\tau)$ denotes the features at time τ. Probabilities for different word sequences given that we have certain observations are expressed with the posterior probability $P(W|\mathbf{O}) \propto P(\mathbf{O}|W)P(W)$, where $P(\mathbf{O}|W)$ is the feature likelihood and $P(W)$ the word sequence probability. Observations are matched against acoustic models to calculate the feature likelihood, and language models are used to obtain the word sequence probabilities. This information is utilised in the decoder to produce the speech recognition result. The speech recognition process is illustrated in Figure 2.1.

In the following sections, we focus on the features and the acoustic models, for speaker segmentation and adaptation are applied on this territory. Features are discussed in Section 2.1 and acoustic models in Section 2.2. Language models are described with a few words in Section 2.3 just in order to complete the picture, and although

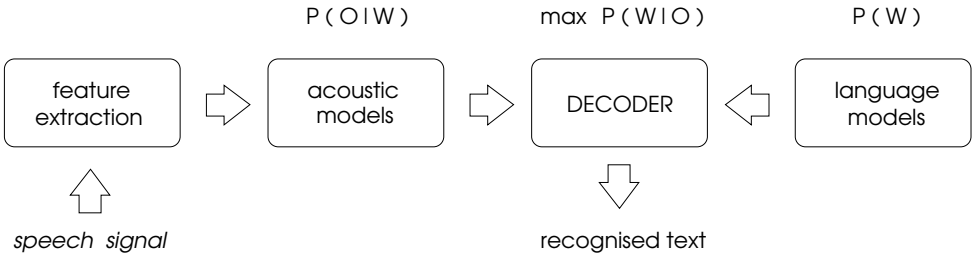

Figure 2.1: Speech signal is represented with features that are linked to letters and words through acoustic models and a lexicon or a pronunciation dictionary. Acoustic information is passed on to the decoder for speech recognition, and language models aid the decoder in constructing meaningful sentences [59].

the decoder is very important in a large vocabulary continuous speech recognition system, it does not need to be discussed within the limits of this work.

2.1 Feature extraction

Speech recognition process begins with speech signal parametrisation. We seek to find features that would preserve the information carried in the speech signal about the words spoken, but at the same time, we wish the features to be invariant to changes in speaker or environment. It is believed that the discriminative information in speech signal is stored where the human ear is most sensitive, for studies on speech and language evolution indicate that our auditory sensitivities have initially restricted the sounds that were considered for speech [39]. The human auditory system along with the speech signal characteristics has given inspiration for several speech modelling techniques like perceptual linear predictive (PLP) [30] and relative spectral (RASTA) [31] analysis.

Speech information is primarily conveyed in the short-time spectrum. Changes in speech signal are relatively slow, so the signal is almost stationary over a short period of time, and can be modelled with a spectrum. Speech signal is hence divided to frames, which are overlapping time windows with duration most often between 10 ms and 25 ms, and a short-time spectrum is calculated from each frame. The spectral information is extracted from the speech waveform with, for example, linear predictive coefficients (LPC) or fast Fourier transform (FFT) and treated with cepstral transformation to separate the harmonic components and the more important vocal tract shape information [26, 59].

Before calculating the short-time spectrum or even dividing the signal to frames, it is filtered with a first-order finite impulse response (FIR) high-pass filter given as

$$H(z) = 1 - az^{-1}, \tag{2.1}$$

where a is typically between 0.9 and 1.0, and values around 0.95 are most popular [59]. Pre-emphasis filtering flattens the spectrum, for speech signal spectrum usually decreases towards the higher frequencies due to physiological characteristics of the human speech production system [17]. Spectrum does not, however, decrease with fricatives like /s/ or /f/. Another motivation for pre-emphasis filtering is that hearing is more sensitive on frequencies above 1 kHz.

For a sampled signal, the spectrum may be calculated with the discrete Fourier transform (DFT), although the computationally more efficient FFT method is most often used in applications. The short-time DFT spectrum in frame τ is calculated as

$$S(k,\tau) = \sum_{n=0}^{N-1} w(n)s(n,\tau)\exp(-j2\pi kn/N) \tag{2.2}$$

where N is the frame length in samples, $s(n,\tau)$ the time-domain speech signal and $w(n)$ a window function. Speech signal is windowed in order to minimise the discontinuities at the frame edges. The standard choice for $w(n)$ is Hamming window [59].

Studies have shown that humans have a good frequency resolution in low frequencies, but distinguishing high frequencies is difficult for us. Thus, a human might say that pitch is halved when the frequency decreases from 200 Hz to 100 Hz, but also when the frequency decreases from 3500 Hz to 1000 Hz. Perceptual frequency scales like mel scale approximate the logarithmic frequency scale of the human ear. To construct a mel-frequency spectrum from the linear FFT spectrum, M triangular bandpass filters are placed onto the frequency axis with uniform spacing in the logarithmic mel scale, and the average energy is computed over each band. Filter bandwidths are such that the width is constant in the logarithmic scale [15].

Finally, a discrete cosine transform is applied to the mel-scale log-energy spectrum to calculate the cepstral features known as mel-frequency cepstral coefficients (MFCC),

$$c(n,\tau) = \frac{2}{M}\sum_{m=1}^{M} \log|P(m,\tau)| \cos\left(m\frac{2\pi}{M}n\right), \tag{2.3}$$

where $P(m,\tau)$ is the average energy calculated from the m-th filter output [17]. The first 10 - 16 MFCC are adopted as features, for they describe the spectral envelope, which in turn characterises vocal tract shape. Spectral envelope can thus discriminate between different phonemes, whereas harmonic components carried in the higher MFCC are not relevant from speech recognition perspective. The other reason for selecting the MFCC features would be that they are almost decorrelated and can be modelled reasonably well with multivariate Gaussian distributions with diagonal covariances. Also, since cepstral transformation separates linear time-invariant (LTI) channel effects and the input signal, removing time-average from the output cepstrum delivers the original speech signal [17].

In addition to the MFCC, logarithmic speech signal power is also used as a feature. The first and second differentials are calculated for the MFCC and power in order to

include temporal dynamics to the features. The delta features are calculated as [59]

$$\Delta(n,\tau) = \sum_{l=1}^{L} l\,(c(n,\tau+l) - c(n,\tau-l))/\sum_{l=1}^{L} l^2, \tag{2.4}$$

where L is the delta-window length in frames. Second differentials or delta-deltas are calculated in replacing the $c(n,\tau)$ with $\Delta(n,\tau)$ in Equation 2.4. MFCC and power features along with their first and second differentials form the feature vectors $\mathbf{o}(\tau)$.

2.2 Acoustic modelling

Phonemes make up spoken words and utterances, just like letters make up written text. In handwritten character recognition, each character is represented with some features and statistical pattern recognition is applied to match the features with a specific letter. We similarly wish to match our features with phonemes. However, phonemes do not correspond to certain features like this, but rather they correspond to feature chains. Phonemes have a time structure, and the acoustic models are needed to account for this. Hidden Markov models (HMM) are a common choice for acoustic modelling, for they are simple, and thus computationally feasible, but still able to catch the important properties of speech quite well [60].

Hidden Markov models [60] are based on Markov chains, statistical models consisting of states and transition probabilities between the states. At any time t, the system is described as being in one of the states $\{1 \dots N\}$. We denote the state at time t as q_t. Given a discrete-time Markov chain, the probability of being in certain state j at the next instant of time is only dependent on the current state i,

$$P(q_{t+1} = j \,|\, q_t = i, q_{t-1} = k, \dots) = P(q_{t+1} = j | q_t = i). \tag{2.5}$$

This probability is called the transition probability from state i to state j and it is often denoted with a_{ij} [60]. Transition probabilities are not dependent on the time t.

The hidden Markov models differ from the classic Markov chain in that the states are not observable, and thus, we cannot see which state the system is in or which states it has visited before. Instead, we get to observe features sampled from state-dependent distributions. Different states can most often generate the same features, but with different probabilities, as illustrated by a simple example in Figure 2.2. The state-dependent probabilities for different features are called observation probabilities. In speech recognition, the states correspond to phonemes or other acoustic units that we wish to recognise, and thus, we would like to find the state sequence hiding behind the features extracted from the speech signal.

We can use the observed features to track which states the system could have travelled if both the transition and the observation probabilities are known. Note that

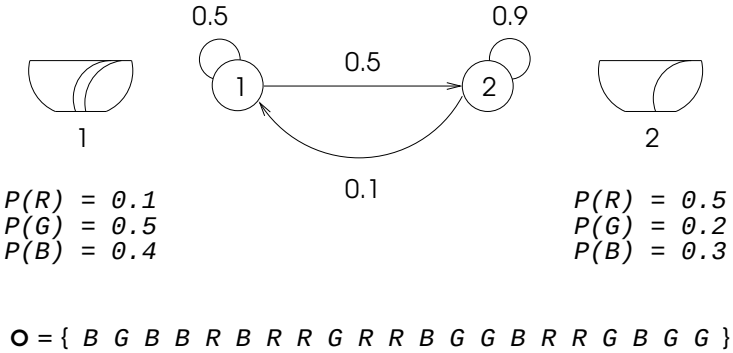

Figure 2.2: Assume there are two large bowls filled with red (*R*), green (*G*) and blue (*B*) balls, and we are given the ratios between different balls in bowls 1 and 2. First, a bowl is selected at random and someone is asked to pick a ball eyes-closed. She will announce which colour ball she picked and return the ball to the container. Then she can either draw another ball from the same bowl or ask for the other bowl and draw a ball from there. If she currently has bowl 1, she is indifferent about this choice, but if she has bowl 2, she is not likely to change the bowl. When she draws a ball from either bowl, we get to hear the colour, but we do not see the bowl nor do we know her decision about changing the bowl. This entire process generates a colour sequence that can be modelled with a two-state hidden Markov model as illustrated above [60].

we cannot just maximise the feature likelihood with respect to the state sequence, for some state sequences that could have well produced the observed features may contain state transitions that are very unlikely. Thus, we seek for the state sequence $\mathbf{q}$ that maximises the joined probability $P(\mathbf{O}, \mathbf{q}|\Lambda)$, where $\mathbf{O} = \{\mathbf{o}(\tau)\}$ are the observed features and Λ the HMM model parameters. This state sequence can be found with the Viterbi algorithm introduced in [70]. Observed features $\mathbf{o}(\tau)$ are assumed statistically independent.

The transition and observations probabilities, HMM model parameters, are learned from data with re-estimation formulas derived from maximising the auxiliary function [60]

$$Q(\Lambda|\Lambda') = \sum_{\mathbf{q}} P(\mathbf{O}, \mathbf{q}|\Lambda') \log P(\mathbf{O}, \mathbf{q}|\Lambda) \tag{2.6}$$

over Λ. The old model parameters are denoted with Λ'. Maximising the auxiliary function is guaranteed not to decrease the feature likelihood $P(\mathbf{O}|\Lambda)$. This optimisation procedure is similar to the EM algorithm with calculating the auxiliary function value given the model set Λ' (expectation) and maximising it over Λ (maximisation).

In order to properly model the continuous observations, HMM states are associated with probability density functions. The most common are the mixture of Gaussians.

With a Gaussian mixture model (GMM), the observation probabilities are given as

$$b_j(\mathbf{o}(\tau)) = \sum_{k=1}^{K} c_{jk} \mathcal{N}(\mathbf{o}(\tau)|\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}), \tag{2.7}$$

where c_{jk} is the weight, $\boldsymbol{\mu}_{jk}$ the mean and $\boldsymbol{\Sigma}_{jk}$ the covariance of the k-th Gaussian in state j [60]. Gaussian mixture models make a reasonable model for the MFCC features often used in speech recognition. Re-estimation formulas for HMM parameters with states modelled as Gaussian mixture models are given in e.g. [60]. Gaussians could also be replaced with another log-concave or elliptically symmetric distribution.

Also, the acoustic units we model need not be phonemes, but other basic speech units could be chosen as well. In fact, phonemes alone are not adequate in representing the spectral and temporal properties of speech, and thus, they are often accompanied with some context information. Triphones, phonemes together with their left and right context, are a common choice for acoustic units. The selected units are modelled as HMM chains with some 3 - 5 states. HMM states may also be associated with duration models [58].

HMM models have served well in speech recognition applications although they are somewhat limited in modelling speech; the Markov assumption itself is inappropriate for one should expect that the dependencies in speech carry further than to the next state alone [60]. An alternative for the HMM based acoustic modelling would be template matching, where acoustic units are associated with a set of possible representations, sequences of speech frames, and pattern matching techniques with dynamic time alignment are then applied to compare the template sequences and the input speech signal [59].

Note that the phoneme or other acoustic unit models are not sufficient in deciphering normal continuous speech, for neighbouring phonemes affect heavily on each other, and thus, recognising a phoneme may be impossible without knowing the context. A phoneme is not necessarily even pronounced, if it is taken to be obvious from the context on word or sentence level. Also, there are no silences in between the words in continuous speech so setting the word boundaries would be impossible without knowing the words.

A lexicon holds the phonetic transcriptions for all the words known to the system. In Finnish, all phonemes apart from /ng/ correspond to a certain letter, but in English, for example, a pronunciation dictionary is needed to map words to phonemes. Lexicon may also contain several transcriptions for a word. With a lexicon, we form word models from the acoustic unit models as illustrated in Figure 2.3. Each word in the lexicon has a unique path connecting the acoustic unit models. Word models are utilised in calculating the feature likelihoods $P(\mathbf{O}|W)$.

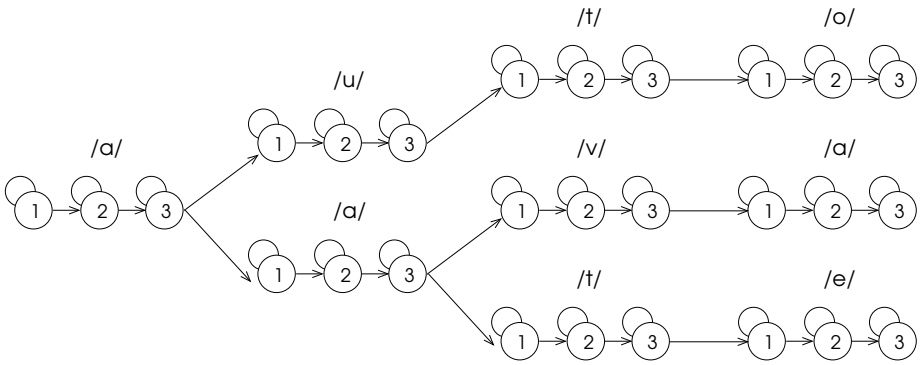

Figure 2.3: Finnish words *auto*, *aava* and *aate* as phoneme sequences with phonemes modelled as three-state HMM chains. Words are stored in a lexicon often represented as a tree where each path from the root to the leaves corresponds to a different word.

2.3 Language modelling

Language models have information of the words and their relations. They can be used to discard improbable words and to decide between acoustically similar words. Suppose, for example, a word has been recognised as being either "hours" or "ours" based on acoustic information and the word models. If we believe the preceding word is "three", the latter interpretation would not make much sense, and the utterance is more likely "three hours".

Language models estimate the word sequence probabilities $P(W)$ most often with an n-gram model. In n-gram modelling, we assume the word probabilities can be estimated based on the $n-1$ preceding words alone. Hence, the word probability is given as $P(w_t) = P(w_t|w_{t-1} \dots w_{t-n+1})$. Word sequence probability, then, is the product of the word probabilities. In most cases, the word probabilities need to be estimated from a textual training corpus by computing the relative frequencies for the word sequences of length n. Relative frequencies calculated for different n may also be interpolated for a more reliable probability estimate [59].

Language models cannot contain all possible words and word forms, for there would be too many. It is customary to choose only the words most frequent in the training corpus to the lexicon. This certainly is a problem in languages like Finnish or Turkish, for example, where inflections are common. Thus, shorter units such as syllables or morphemes would be more suited than words for language modelling [32]. An unsupervised method for extracting morpheme-like subword units from text corpus is presented in [14]. A corpus segmented to morphs may be used to train morph-based language models, that are expected to cover the language better than models using the same number of words could do. It should be noted, though, that word boundaries are not recognised automatically as with words, but the recogniser needs to hypothesise a word boundary after each morph.

Chapter 3

Speaker-Based Adaptation

Variation in speaker characteristics as well as in environmental conditions degrade the speech recognition performance as the acoustic models become mismatched with the speech data. Speaker-based adaptation and normalisation seek to compensate the mismatch between the speaker-independent model and the speech data. Speaker adaptation methods work in the model-space transforming the acoustic models to better match the new speaker and conditions, whereas speaker normalisation methods modulate the speech signal for a better resemblance with the average speaker represented by the acoustic models.

The most common method for speaker-based normalisation is vocal tract length normalisation (VTLN), which is used to compensate the variation different vocal tract lengths introduce to speech spectrum (see Figure 3.1). VTLN methods re-scale the frequency axis to normalise the spectrum. There is usually one parameter that defines how the frequency axis should be scaled. This may be estimated based on formant positions [20], but it is more common to perform a grid search over possible parameter values and choose the frequency warping that maximises the likelihood of normalised features [74]. VTLN methods do not generally produce error reductions quite as notable as speaker adaptation methods [75], but if normalisation is applied to the acoustic training data, VTLN methods can substantially improve the recognition results when the amount of training data is very limited [20].

Speaker adaptation methods come in many forms, but most common so far have been the maximum a posteriori (MAP) methods and linear transformation methods such as maximum likelihood linear regression (MLLR). Also, there are some interesting speaker clustering or speaker space methods like cluster adaptive training (CAT) [22] and eigenvoice techniques [40]. The general idea is to combine the speaker-dependent models for the unknown test speakers from a set of canonical models, that have been estimated from the training data. The speaker clustering/speaker space adaptation methods are strong in that they can give a good performance on small amount of adaptation data [72]. However, if the environmental conditions are considerably different in training and test data, the canonical speaker models need to be adapted

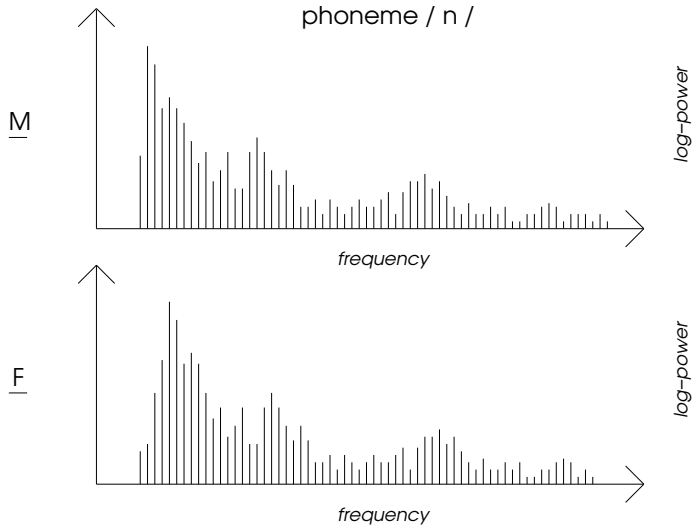

Figure 3.1: Women have on average a shorter vocal tract than men, and thus, the formants in speech signal spectrum are located at higher frequencies when speaker is female. Female speaker may exhibit formants even 20 % higher compared to male speakers [20]. In order to normalise speech to correspond to the speaker-independent models trained with both male and female speakers, female speech spectra are most often compressed and male speech spectra are stretched.

to match the test environment in order to get proper results [52].

It is not uncommon to combine speaker normalisation and adaptation, either. Error reductions are essentially additive when maximum likelihood linear regression is applied after vocal tract length normalisation [56, 69, 75]. However, VTLN methods do not improve the results as much if followed with constrained maximum likelihood linear regression (CMLLR) instead [69]. CMLLR differs from MLLR in that linear transformations are applied to features rather than acoustic model parameters. Pitz and Ney [55] proved analytically that frequency warping corresponds to linear transformation in cepstral space, and thus, to constrained maximum likelihood linear regression with the restriction to one adjustable parameter. The result was derived with plain cepstrum, but is expected to hold for mel-warped cepstrum alike.

Speaker-based adaptation has also been researched in the Laboratory of Computer and Information Science. Siivola [63] introduced an online speaker adaptation method derived from maximum a posteriori adaptation and self-organising maps (SOM). Creutz [13] experimented with MAP and MLLR adaptation, and proposed a method for duration adaptation to compensate for variations in speech rate. A method for selecting adaptation data with certain phonetical profile was also developed and tested.

In this work, we concentrate on the linear transformation methods and the constrained maximum likelihood linear regression (CMLLR) in particular, but also

MAP and MLLR are discussed briefly. Speaker adaptation methods are assessed based on their needs for adaptation data and their convergence properties, for ideally, we would be able to get a good estimate for the speaker adapted model based on a limited amount of adaptation data, and should more data become available, the adapted model would converge to the true speaker-dependent model. The true speaker-dependent model is the model we would get if we trained the acoustic models with speech data from this specific speaker.

3.1 Maximum a posteriori estimation

In maximum a posteriori (MAP) parameter estimation, generally, the parameter values are chosen from the mode of the distribution $p(X|\Lambda)p(\Lambda)$, where X is the observed data and Λ denotes the model parameters. With prior distribution $p(\Lambda)$, not much data is needed to get valid parameter estimates. Thus, MAP estimation would seem well suited for speaker adaptation purposes.

The prior distribution is commonly chosen from the same family as the posterior distribution. However, for mixture Gaussian HMM models such conjugate prior does not exist in finite dimension. Gauvain and Lee [25] thus proposed on alternative approach where the GMM means are updated as

$$\hat{\boldsymbol{\mu}}_k = \frac{\upsilon\, \boldsymbol{\mu}_k^{(0)} + \sum_\tau \gamma_k(\tau)\mathbf{o}(\tau)}{\upsilon + \sum_\tau \gamma_k(\tau)} \tag{3.1}$$

where $\hat{\boldsymbol{\mu}}_k$ is the adapted mean for Gaussian k and $\boldsymbol{\mu}_k^{(0)}$ the prior mean value, $\gamma_k(\tau)$ are the posterior probabilities for being in Gaussian k at time τ given the observed features $\mathbf{o}(\tau)$, and υ is a meta-parameter that gives the bias between the observation-based maximum likelihood mean estimate and the prior mean [72]. Similar rules may be used to update the mixture variances and weights [25].

As the amount of adaptation data increases towards infinity, the MAP estimates converge to the maximum likelihood estimates and the adapted model becomes similar with the true speaker-dependent model [25]. When the adaptation data is limited in size, however, problems arise from that the MAP approach is a local approach in the sense that only parameters that are observed in the adaptation data are updated. Thus, the adaptation rate is usually slow, and the models adapted based on one short utterance from a specific speaker do not necessarily make any difference in recognising other utterances from the same speaker.

3.2 Linear transformation adaptation

Model parameters may also be adapted with a linear transformation. The advantage, here, is that the same transformation may be used on several, if not all, model

parameters, so that even the parameters not observed in the adaptation data may be adapted to the new speaker. Regression classes, that share the transformation, may be defined based on broad phonetic classes or clustering similar mixture components with a regression class tree method [42].

Legletter and Woodland [43] proposed the maximum likelihood linear regression (MLLR) for Gaussian mean parameter adaptation, and Gales and Woodland [23] further extended the MLLR framework to adapting the model covariances. In MLLR, the Gaussian mean parameters are updated according to a linear transformation as

$$\hat{\boldsymbol{\mu}} = \mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \tag{3.2}$$

where $\mathbf{A}$ and $\mathbf{b}$ are the mean transformation matrix and bias. If both the means and covariances are adapted, it is common to optimise the transformation parameters in two stages. Thus, we first keep the covariances fixed and estimate the mean transformation parameters so that the adapted models maximise the feature likelihood in the adaptation data set. Then we trade places and fix the mean and find the maximum likelihood transformation for the covariances. The covariances $\boldsymbol{\Sigma}$ are updated as [23]

$$\hat{\boldsymbol{\Sigma}} = \mathbf{L}\mathbf{H}\mathbf{L}^T, \tag{3.3}$$

where $\mathbf{L}$ is the Choleski factor of the covariance matrix $\boldsymbol{\Sigma}$ and $\mathbf{H}$ is the covariance transformation matrix, or with an alternative rule proposed in [21],

$$\hat{\boldsymbol{\Sigma}} = \mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^T. \tag{3.4}$$

When $\mathbf{H}$ is a simple diagonal transformation, the same results are obtained with both approaches, but with full transformations, the transformation matrices as well as the final results depend on whether Equation 3.3 or Equation 3.4 is chosen [21].

Linear transformation methods usually outperform the MAP adaptation when relatively small amount of data is available for parameter estimation. However, linear transformation methods also do not benefit from more adaptation data after certain limit, and thus, the speaker-dependent performance may never be reached. This would, of course, depend on how many different transformations are estimated for the model [72].

MAP and MLLR adaptation are both applied to the model parameters directly. Thus, in speaker-based adaptation, the methods create a new model for each speaker. A transformation applied to features rather than model parameters would be better in a multi-speaker environment, since we then needed to store just the speaker-specific transformations, and our memory requirements would be significantly relieved [45]. Feature transformations are also more convenient than model transformations in speaker adaptive training (SAT). In this technique, the HMM parameters and the speaker-specific transformations are both estimated in model training, and the acoustic models are expected to become less dependent on the speaker specific characteristics than traditional speaker-independent models [50].

Digalakis et al. [19] constrained the transformation so that the models are updated as

$$\hat{\boldsymbol{\mu}} = \mathbf{A}'\boldsymbol{\mu} + \mathbf{b}' \tag{3.5}$$

$$\hat{\boldsymbol{\Sigma}} = \mathbf{A}'\boldsymbol{\Sigma}\mathbf{A}'^{T}, \tag{3.6}$$

where $\mathbf{A}'$ and $\mathbf{b}'$ are the transformation matrix and bias for the constrained maximum likelihood linear regression (CMLLR). CMLLR is fundamentally a model-space transformation, but in practice, adaptation can be done in the feature space and the original model parameters need not to be touched.

To rewrite the model-space transformation as a feature-space transformation, remember that we wish to set the transformation parameters so that the transformed model parameters would maximise the feature likelihood. In practice, we are to maximise the auxiliary function $Q(\Lambda|\Lambda')$ which becomes

$$Q(\Lambda|\Lambda') = K - \frac{1}{2}\sum_{k=1}^{K}\sum_{\tau=1}^{T}\gamma_k(\tau)[K_k + \log(|\hat{\boldsymbol{\Sigma}}_k|) + (\mathbf{o}(\tau) - \hat{\boldsymbol{\mu}}_k)^T\hat{\boldsymbol{\Sigma}}_k^{-1}(\mathbf{o}(\tau) - \hat{\boldsymbol{\mu}}_k)], \tag{3.7}$$

where the model parameters Λ and Λ' include the linear transformation matrix $\mathbf{A}'$ and bias $\mathbf{b}'$ and the Gaussian means and covariances. The transformed mean and covariance in Gaussian k are denoted with $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\Sigma}}_k$, and $\gamma_k(\tau)$ are the posterior probabilities of being in Gaussian k at time τ given the observed features $\mathbf{o}(\tau)$. K is a constant dependent only on the HMM transition probabilities, and K_k is the normalisation constant associated with Gaussian k [21]. If we substitute Equation 3.5 and Equation 3.6 in Equation 3.7 and rearrange, we have

$$Q(\Lambda|\Lambda') = K - \frac{1}{2}\sum_{k=1}^{K}\sum_{\tau=1}^{T}\gamma_k(\tau)[K_k + \log(|\boldsymbol{\Sigma}_k|) - \log(|\mathbf{A}|^2) + (\hat{\mathbf{o}}(\tau) - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\hat{\mathbf{o}}(\tau) - \boldsymbol{\mu}_k)], \tag{3.8}$$

where the model transformation has been written as a feature transformation with

$$\hat{\mathbf{o}}(\tau) = \mathbf{A}\mathbf{o}(\tau) + \mathbf{b} = \mathbf{W}\boldsymbol{\zeta}(\tau), \tag{3.9}$$

where the transformation matrix for features $\mathbf{A} = \mathbf{A}'^{-1}$ and bias $\mathbf{b} = \mathbf{A}'^{-1}\mathbf{b}'$, and $\mathbf{W}$ is the extended transformation matrix and $\boldsymbol{\zeta}(\tau)$ is the extended observation vector at time τ [21]. CMLLR is not strictly feature-space because the transformation also affects likelihood calculations. Feature log-likelihoods have to be corrected by $\frac{1}{2}\ln|\mathbf{A}|^2$ if likelihoods need to be comparable between features not adapted similarly.

We wish to set the linear transformation parameters so that the transformed features maximise the auxiliary function. Digalakis et al. [19] introduced the maximum likelihood solution for a diagonal feature transformation matrix $\mathbf{A}$, and Gales [21] later extended the CMLLR estimation to cover full scaling matrices. Estimation is again based on the expectation-maximisation (EM) algorithm [18] and requires iterative optimisation over rows. The maximum likelihood estimate for the i-th row in $\mathbf{W}$ is given as

$$\mathbf{w}_i = (\alpha\mathbf{p}_i + \mathbf{k}(i))\mathbf{G}(i)^{-1} \tag{3.10}$$

where $\mathbf{p}_i$ is the extended cofactor vector $\mathbf{p}_i = [\,0\, c_{i1} \ldots c_{in}]$ and $c_{ij} = \mathrm{cof}(\mathbf{A}_{ij})$, and n the feature dimension. Factor α is solved from a simple quadratic equation presented in [21]. Statistics $\mathbf{G}(i)$ and $\mathbf{k}(i)$ are calculated as

$$\mathbf{G}(i) = \sum_{\tau=1}^{T} \boldsymbol{\zeta}(\tau)\boldsymbol{\zeta}(\tau)^T \sum_{k=1}^{K} \sigma_k(i)^{-2}\gamma_k(\tau) \tag{3.11}$$

$$\mathbf{k}(i) = \sum_{\tau=1}^{T} \boldsymbol{\zeta}(\tau)^T \sum_{k=1}^{K} \sigma_k(i)^{-2}\mu_k(i)\,\gamma_k(\tau), \tag{3.12}$$

where $\mu_k(i)$ and $\sigma_k(i)^2$ denote the i-th components of the mean and variance of Gaussian k and, again, $\gamma_k(\tau)$ are the posterior probabilities of being in Gaussian k at time τ. Here, the model covariances are assumed diagonal, but similar estimation formulas are derived for full-covariance models in [65]. Note, that $\mathbf{G}(i)$ and $\mathbf{k}(i)$ are summations over a time interval, and thus, they may be calculated incrementally as more data becomes available.

3.3 Robustness in linear transformation methods

In order to calculate the posterior probabilities for the Gaussians, we need to have the state sequence that produced the observed features, or consistently, the words that were spoken. If we indeed have a word-level transcription of the adaptation data, the adaptation is supervised. In unsupervised adaptation, then, the true words uttered are not known, and we must use a decoder given hypothesis instead. Linear transformations are well-suited for unsupervised adaptation since many parameters share the same transformation, and thus, occasional errors in the transcription are averaged out.

Linear transformations also generalise well on unseen data from the same speaker and conditions. However, if not enough data is provided for parameter estimation, the results are unreliable and may increase the error rates compared to the baseline [72]. To ensure that the performance is never poorer than with the speaker-independent model, a threshold needs to be set to prevent speaker adaptation from being used if enough data is not available. The threshold might be some 1000-1500 frames (around 10 seconds) for estimating a full scaling matrix from 39 dimensional observations [72]. This corresponds on average to two spoken sentences. For diagonal or block-diagonal matrices, less data is needed.

Various solutions have been proposed for making the linear transformation based speaker adaptation feasible also when the amount of adaptation data is severely limited. Chesta et al. [10] introduced a Bayesian counterpart for the standard MLLR, the maximum a posteriori linear regression (MAPLR). Gunawardana and Byrne [29] proposed discounted likelihood estimation procedure for the linear transformation

parameters. In discounted likelihood linear regression (DLLR), the adaptation statistics are interpolated with the speaker independent statistics to avoid overtraining.

Lei et al. [44] introduced a feature-space equivalent for MAPLR, where the MAP estimate for $\mathbf{W}$ is calculated according to Equation 3.10 from the smoothed statistics

$$\tilde{\mathbf{G}}(i) = \mathbf{G}(i) + \mathbf{S}(i)^{-1} \tag{3.13}$$

$$\tilde{\mathbf{k}}(i) = \mathbf{k}(i) + \mathbf{S}(i)^{-1}\mathbf{m}(i), \tag{3.14}$$

where $\mathbf{G}(i)$ and $\mathbf{k}(i)$ are as defined in Equations 3.11 and 3.12, and $\mathbf{m}(i)$ and $\mathbf{S}(i)$ are the prior mean and covariance for the i-th row in $\mathbf{W}$. In Bayesian estimation, the prior is assumed to be known, but we may take the empirical Bayes approach [61] and estimate the prior from data. Setting the prior is still a problem, since we might not have enough data similar to the test data to estimate the prior from.

Chapter 4

Speaker-Based Segmentation

Speaker-based segmentation aims at dividing the input audio to personalised speaker turns. Given no prior information on speakers, the task is to find where speaker changes take place and then label the detected speaker turns as illustrated in Figure 4.1. Speakers are not connected to any certain label, but the labels should be assigned so that the speaker turns with the same speaker share a common label, and this label is not assigned to any speaker turns from another speaker. In other words, the speaker turns should be clustered so that each cluster corresponds to a single speaker.

Numerous speaker change detection (SCD) methods have been proposed for finding the speaker change boundaries. In this work, the focus is on metric-based speaker change detection, where the speaker change boundaries are found using a distance measure that illustrates the dissimilarity between two speech segments. Model-based speaker change detection methods would use for example Gaussian mixture models trained for different acoustic classes and assign speech segments to the classes according to maximum likelihood principle. Speaker change boundaries would then be assumed where a change in the acoustic class occurs. Speaker change boundaries may also be placed where there is silence in the speech stream. Silences are detected either by the decoder or directly by measuring and thresholding the energy of the audio signal [38]. Additionally, Canseco-Rodriguez et al. [8] proposed using linguistic

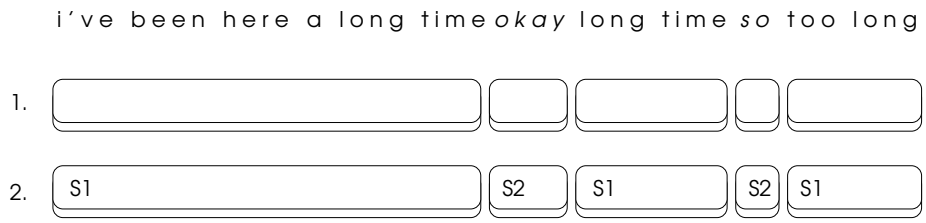

Figure 4.1: Speaker segmentation first divides the input audio to speaker turns and then labels the detected turns. The above dialog is taken from the Fisher corpus [11].

information for speaker change detection in broadcast news transcription task where certain phrases are common when reporters begin and end their stories.

An alternative for speaker change detection would be to partition the audio into equally-length speech segments and label them as we would label the speaker turns. Speaker change boundaries are then set where the speaker label changes. To make sure there is speech from one speaker only in each segment, the segments have to be made very short. The segment length is typically set between 0.25 and 2 seconds. Speaker change detection should create longer segments and thus provide more data for labelling [45].

Relation between speaker recognition and labelling speaker turns without any prior information on speakers is similar to that between general classification and clustering tasks. In speaker recognition, we would train a model for each speaker and use the speaker models to decide who spoke the given utterance. This corresponds to a classification task. When speakers are not known beforehand, the classification is to be done unsupervised: speaker turns are clustered based on some similarity measure and the clusters thus discovered are adopted as classes. They are associated with unique class labels, and the speaker turns are then labelled according to the cluster they belong to. Each cluster is assumed to correspond to a separate speaker, and class labels are thus referred to as speaker labels in the following discussions.

We will discuss two methods for speaker turn clustering. In speaker model based clustering, each speaker turn is associated with a feature distribution model. It is assumed, that the models are different for different speaker, but quite similar for speaker turns that share the same speakers. Thus, speaker turns may be clustered according to the distance or dissimilarity between their respective models.

The other option we are interested in is based on the assumption that a speaker-specific transformation estimated for speaker adaptation purposes is unique in the sense that each speaker has an optimal transformation, and this transformation is not optimal for any other speaker. Thus, if we estimate a transformation for a specific speaker, no other speaker can benefit from this transformation as much as he would benefit from a transformation estimated specifically for him. Provided that we get reliable estimates for the speaker-specific transformations, we can recognise different speakers based on these transformations. Transformations can be used as speaker models [67] or speaker turns can be clustered so that the clustering solution may be expected to maximise the feature likelihood as the features are adapted with speaker-specific transformations estimated for each cluster separately [37].

In the following sections we discuss some methods commonly used in speaker-based segmentation. Section 4.1 concerns some general issues related to metric-based speaker change detection methods and presents common distance measures for speaker model comparison. Section 4.2 discusses speaker model based clustering, and in Section 4.3 speaker turn clustering is approached form the speaker adaptation perspective. Other speaker turn clustering methods such as spectral clustering [53] still

remain outside the scope of this work. Also, emphasis is on methods that can be applied to online clustering. Offline methods assume that all the data that is to be clustered is available from the beginning, but in online clustering, the speaker turns are clustered sequentially, and any information about the forthcoming speaker turns is not used in decision making.

4.1 Speaker change detection

Features extracted from speech signal characterise both the spoken message and the speaker and acoustic conditions. However, features collected from more than few seconds of speech are expected to fill the feature space in a way that depends primarily on the speaker and acoustic conditions, and not the particular text spoken [26]. Thus, if there are no changes in acoustic conditions, speaker change at time t can be detected comparing feature sets $X = \{\mathbf{x}_n\}$ and $Y = \{\mathbf{y}_m\}$ that correspond to speech uttered before and after time t.

The feature sets are often marked with moving windows placed at time t and shifted with a fixed step Δt that determines the resolution of detected speaker turns [16, 38, 48]. As an alternative, Liu and Kubala [46] suggested testing for speaker change at hypothesised phone boundaries. The phone level time resolution makes speaker change detection less time consuming while maintaining an accuracy comparable to the frame level approach. Hypothesised phone boundaries may not be accurate, but in speaker adaptation, the transformations are estimated based on phonetic transcriptions, and thus, even if the speaker change boundary had been found at a frame that does not correspond to a hypothesised phone boundary, it would likely be moved to the next phone boundary location when the speaker-specific transformations are estimated. The phone level approach with moving windows is illustrated in Figure 4.2.

Another common approach uses a variable-size increasing window to mark a search area that should contain one speaker change at the most. To test for speaker change, the search area is bisected at time t to make the feature sets X and Y (Figure 4.3). Speaker changes are tested at evenly spaced time intervals or at the hypothesised phone boundaries, for example. The window size is gradually increased and testing is repeated inside the window until a speaker change is found. Then, a new window is initialized and set to begin at the detected speaker change boundary. A more detailed description is presented in [9].

With the variable-size window we get as much features as possible for making the final decision, but we also test for speaker change at the same locations several times during the search and the method is thus more time consuming than the moving windows approach. Also, the window size must not increase too much in one step in order to avoid having more than one speaker change boundary inside the search area. Tritschler and Gopinath [68] proposed a rule that increases the window size

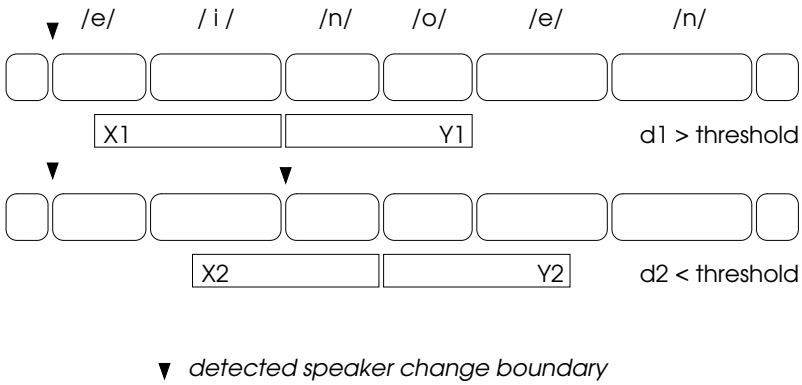

Figure 4.2: Speaker change is tested at consecutive hypothesised phone boundaries with moving windows. If distance between the features X and Y passes a certain threshold, we conclude there is a speaker change boundary in between the windows.

more when speaker change boundaries are not likely to occur and suggested not to re-calculate the distances at the first possible speaker change boundary locations inside the current window as it is very unlikely that a speaker change boundary should be found at this area. Zdansky [73] later introduced some modifications that make it possible to have multiple speaker change boundaries inside the search area.

Delacourt and Wellekens [16] proposed a two-step speaker change detection scheme that first uses the moving windows approach for initial speaker change detection (step 1) and then takes all the features from inside the detected speaker turns as feature sets X and Y in order to validate the detected speaker change boundaries (step 2). This second pass is expected to remove the many false boundaries that speaker change detection might create. The procedure is often referred to as speaker change boundary refinement [48] or local clustering [53].

The speaker change boundaries are observed as local maxima at the trajectory of the chosen distance or dissimilarity measure $d = d(t)$. Significant local maxima are detected in comparing the distance to a threshold value T. Liu and Kubala [46] segment the audio to speech and non-speech before applying speaker change detection, and since most speaker changes are expected to happen at non-speech, they set the threshold higher for speech frames. Lu and Zhang [48] propose an adaptive threshold for speaker change detection under varying environmental conditions, and Delacourt and Wellekens [16] choose the significant local maxima based on local comparisons rather than the absolute distance.

In an application where speaker turns are clustered after speaker change detection, missing true speaker change boundaries is more severe than detecting false boundaries, for clustering can remove spurious speaker turn boundaries, whereas missed boundaries cannot be recovered. Missing true boundaries also leaves speech data from two or more speakers inside one speaker turn and degrades clustering performance. It is thus customary in speaker segmentation task to set the threshold in

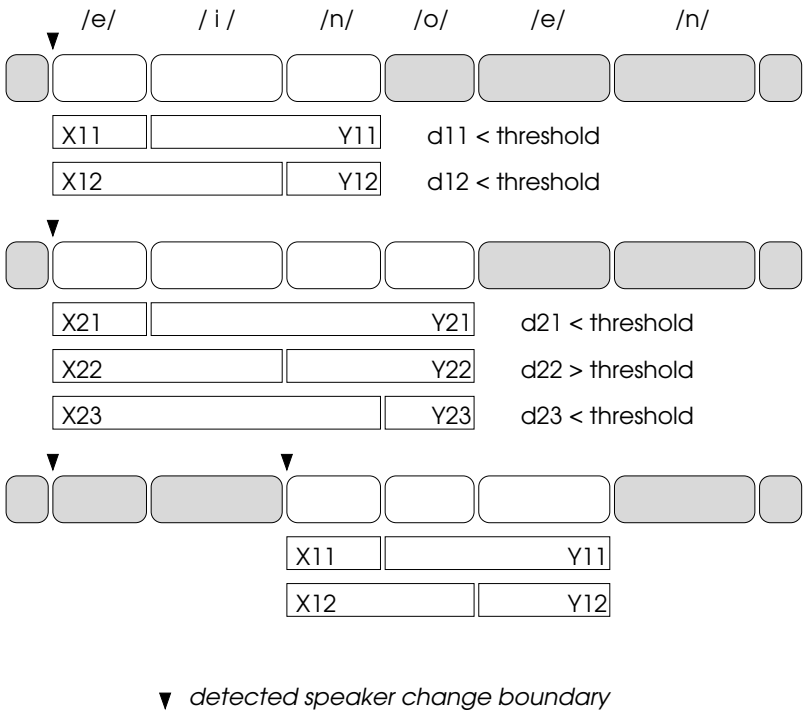

Figure 4.3: Speaker change is tested at each hypothesised phone boundary inside the current window. The window is divided to features sets X and Y at the hypothesised phone boundaries. Speaker change is detected when the distance between the feature sets passes a certain threshold. If no speaker change is found, the window size is increased and testing is repeated at every hypothesised phone boundary inside the window. If, then, a speaker change is detected, we set a new window to begin from the speaker change boundary. In case there are multiple hypothesised phone boundaries where the distance between feature sets passes the threshold, speaker change boundary is placed where the distance is greatest.

speaker change detection so that false detections are more likely than missing true boundaries [68].

Various distance measures have been used to compare the feature sets and detect speaker changes. The most common choices would be the generalised likelihood ratio (GLR) [27, 47, 36] and the Bayesian information criterion (BIC) [9, 53, 68]. These will be discussed in further detail along with another popular distance measure, the Kullback-Leibler (KL) divergence [48, 7, 53, 62]. Other possibilities would include the entropy loss tested in [38] and the similarity measures introduced for speaker recognition in [6] and later applied to speaker change detection task in [16].

Generalised likelihood ratio

To determine if feature sets $X = \{\mathbf{x}_n\}$ and $Y = \{\mathbf{y}_m\}$ represent the same speaker, we assume the features are statistically independent and can be modelled as coming from a multivariate Gaussian distribution. The question is, now, whether or not the features are from the same Gaussian distribution, and we wish to test the hypothesis:

H_0: X and Y are generated by the same speaker

H_1: X and Y are generated by different speakers

The likelihood that features in X and Y are from the same Gaussian distribution and thus represent the same speaker is $L_0 = p(X, Y|\Lambda)$ and the likelihood that the features come from different distributions is $L_1 = p(X|\Lambda_1)\, p(Y|\Lambda_2)$, where Λ_1 and Λ_2 denote the model parameters. Taking the common likelihood ratio test for testing that several normal distributions are identical and replacing the unknown model parameters with their maximum likelihood estimates we arrive at the generalised likelihood ratio (GLR) test. The test criterion is

$$R = \frac{\max L_0}{\max L_1} \tag{4.1}$$

with the maximum likelihood values calculated as $\max p(X|\Lambda) = [\ (2\pi e)^p\ |\hat{\mathbf{\Sigma}}|\ ]^{-\frac{1}{2}N}$, where p is the feature dimension, N the feature set size and $\hat{\mathbf{\Sigma}}$ the maximum likelihood estimate calculated for the model covariance [5]. Note that $\max L_1 \geq \max L_0$, and the equality holds when the maximum likelihood model estimates for the feature sets X and Y are exactly the same. This cannot happen unless the true probability distributions are Gaussians, which is not generally true for features used in speech recognition [4].

The generalised likelihood ratio is always greater than zero and less than unity. Thus, we may define a distance measure comparable to the generalised likelihood ratio by taking the negative of its logarithm [27]. The generalised likelihood ratio distance is calculated as

$$d_{GLR} = -\frac{1}{2}\left[N \log |\mathbf{S}_1| + M \log |\mathbf{S}_2| - (N+M) \log |\mathbf{S}|\right], \tag{4.2}$$

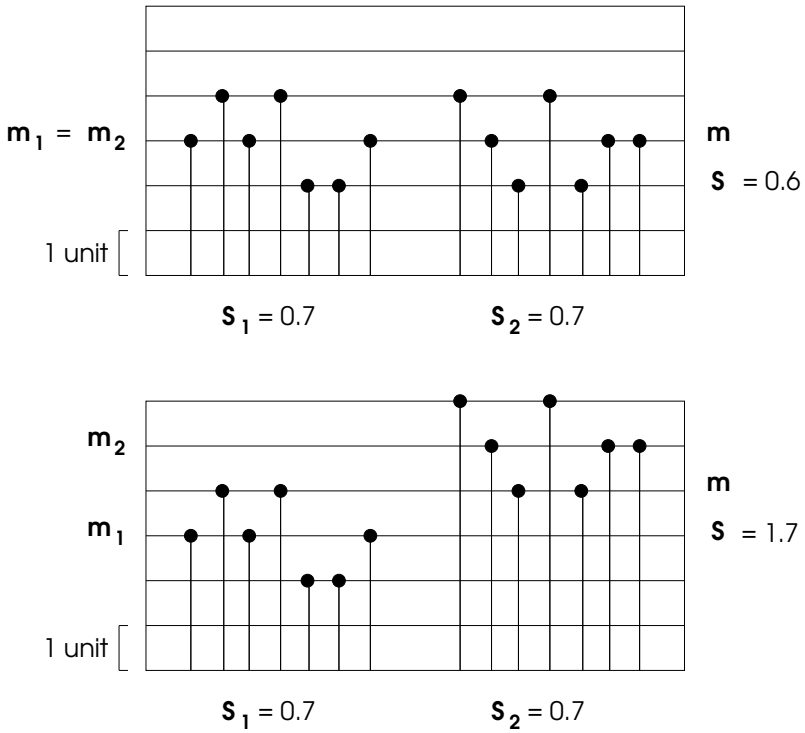

Figure 4.4: Assume we have two one-dimensional feature sets with identical sample covariances $\mathbf{S}_1$ and $\mathbf{S}_2$. If the feature sets have the same mean, the sample covariance $\mathbf{S}$ calculated over both feature sets is close to the covariances $\mathbf{S}_1$ and $\mathbf{S}_2$, but if the mean is different in the two feature sets, the sample covariance $\mathbf{S}$ becomes greater. Thus, the generalised likelihood ratio distance which is based on comparisons between the sample covariances is also dependent on the differences observed in the mean value. As the mean is more sensitive to various environmental conditions than the covariance, distance measures that do not depend on the mean estimates are sometimes preferred [27, 48].

where $\mathbf{S}_1$ is the sample covariance matrix calculated from features X, $\mathbf{S}_2$ the sample covariance matrix calculated from features Y, and $\mathbf{S}$ the sample covariance matrix calculated from features $X \cup Y$. N, M denote the number of features in X and Y, respectively. Note that the distance depends on both the mean and covariance estimates (see Figure 4.4). To test the similarity between the covariances only, the sample covariance matrix $\mathbf{S}$ should be replaced with $(N\,\mathbf{S}_1 + M\,\mathbf{S}_2)/(N+M)$ [27].

To determine if the feature sets represent the same speaker, we compare the distance to a threshold. Threshold T is selected so that when the distance is over the threshold, there may be a speaker change in between the feature sets. The threshold can be found experimentally, or it can be determined based on information theoretical measures such as minimum description length (MDL) or Bayesian information criterion (BIC).

Gish et al. [27] noticed that the generalised likelihood ratio and thus the proposed

distance measure are dependent on the speech segments length. They proved experimentally that distance between the same two speakers was measured greater when more data was available. This can be explained as resulting from not having enough data to reach the asymptotic region where the distribution would not change, and also from that Gaussian model is an approximation of the true feature distribution. Thus, if we wish to have a static threshold for the distance, we have to keep the window size constant. Gish et al. [27] also observed that the GLR distance measure provides a better discrimination among speakers if both speech segments are similar in length.

If one should insist on using the variable-size window scheme, the distance should be explicitly made dependent on the number of features in each feature set. Liu and Kubala [46] propose $\theta \log(N + M)$, where θ is set experimentally, to be subtracted from the distance. This is similar to the BIC threshold proposed in [9].

Bayesian information criterion

The likelihood that the features in X and Y represent the same speaker is effectively a measure of how well a Gaussian model can explain the observations, whereas the likelihood that we have different speakers is comparable to how well a model with two Gaussian components can perform this. This favours the interpretation that feature sets are not from the same speaker, for the model with two Gaussians is clearly more complex.

Bayesian information criterion introduces a complexity penalty to the model likelihood. With N features and K model parameters the BIC complexity penalty for log-likelihood is $\frac{1}{2}K \log N$. The BIC defined threshold for speaker change detection would thus be $\frac{1}{2}\Delta K \log N$. The model for different speakers has one Gaussian more than the model for one speaker, so $\Delta K = p + \frac{1}{2}p(p + 1)$, where p is the feature dimension and we assume full covariance matrices are estimated for the Gaussians.

GLR distance and the threshold are often presented as the BIC distance measure [9]

$$d_{BIC} = d_{GLR} - \frac{1}{2}(p + \frac{1}{2}\, p\,(p + 1)) \log(N + M)), \tag{4.3}$$

where p is the feature dimension and N, M the number of features in X and Y. This has the decision boundary at zero and no separate threshold is needed. However, the results are often better when we have the GLR distance with a carefully chosen threshold [38]. Also, a common approach is to include scaling parameter λ in the complexity penalty term [4, 16, 68]. Adjusting the scale is essentially equivalent to adjusting the GLR distance threshold.

GLR distance and BIC distance are the most commonly used criteria for speaker change detection. Although BIC distance does not need a threshold, it is often equipped with a scaling parameter at least. Thresholds and scaling parameters are

very sensitive to changes in the acoustic conditions and have to be tuned for the new audio each time. An alternative proposed in [4] is to have a GMM with two mixture components to model the feature sets X and Y when they are assumed to have a common distribution. The models then have the same complexity and the likelihood comparison favours neither model over another, so no threshold is needed.

BIC distance is a little sensitive to not having enough data for parameter estimation. Thus, it may be reasonable not to use the BIC criterion in the first pass, but rather detect speaker changes with some other measure first and use the BIC distance for local clustering [16].

Kullback-Leibler divergence

To determine whether the feature sets X and Y represent the same speaker or different speakers, we have estimated models for both options and compared the likelihoods. Another option would be to directly compare the models estimated for the feature sets: if the models for X and Y are similar, we may assume the features were generated by the same speaker, whereas gross differences in the models would indicate a speaker change.

Distance measures for speaker models are often derived from the Kullback-Leibler divergence [48, 7, 53, 62]. Other possibilities for Gaussian features would include the Mahalanobis distance and the Bhattacharyya distance [7]. The Kullback-Leibler divergence measures the distance from the true probability distribution $f(x)$ to some probability distribution $g(x)$ as [12]

$$KL(f||g) = \int f(x) \left[\log f(x) - \log g(x)\right] dx. \tag{4.4}$$

This may be seen as the expected log-likelihood ratio for testing the hypothesis that $f(x)$ should be chosen over $g(x)$ to model the features x, when indeed $f(x)$ is the correct model. The Kullback-Leibler divergence is not symmetric. The distance is measured from one model to the other, not between the models. Siegler et al. [62] thus defined the Kullback-Leibler distance measure for speaker segmentation with the symmetrised divergence as $d_{KL} = KL(f||g) + KL(g||f)$.

With Gaussian features, the KL distance between the feature sets is calculated as [7]

$$d_{KL} = \frac{1}{2}\mathrm{tr}[(\mathbf{S}_1 - \mathbf{S}_2)(\mathbf{S}_2^{-1} - \mathbf{S}_1^{-1})] + \frac{1}{2}\mathrm{tr}[(\mathbf{S}_1^{-1} + \mathbf{S}_2^{-1})\delta\delta^T], \tag{4.5}$$

where $\mathbf{S}_1$ and $\mathbf{S}_2$ are the sample covariance matrices calculated from the features and δ is the difference between the sample mean vectors, $\delta = (\mathbf{m}_1 - \mathbf{m}_2)$. The distance $d_{KL} \geq 0$ and the equality holds when the models are exactly the same, which is not possible unless the true feature distributions are Gaussians. The features used in speech recognition are not generally Gaussian distributed [4].

The first component on the right side of the Equation 4.5 may be characterised as the measure of differences in shape and the second the measure of differences in size of the probability distributions. The shape component, or divergence shape distance, is a common choice for distance measure in speaker change detection and clustering, for it does not depend on the estimated means that may be biased due to various environmental conditions [7, 48].

4.2 Speaker model based clustering

Speaker turn clustering addresses the problem of grouping the unlabelled speaker turns so that utterances from one speaker all are in the same group and utterances from different speakers are all in separate groups. In speaker model based clustering the speaker turns are represented with a model, most often a Gaussian or a GMM. Similarity between two models is measured with the same metrics that are used in speaker change detection.

With a similarity measure for the speaker models defined, all common clustering approaches like hierarchical clustering or k-means clustering may be used in finding the groups of speaker turns corresponding to one speaker. In hierarchical clustering we sequentially find the two models closest to each other and merge them. We can either recalculate the model parameters using all the features assigned to the cluster, or we can update the distance between this cluster and the other clusters or models according to certain rules.

The most important question is, then, how to choose the number of clusters, as we have no prior information on speakers. The hierarchical clustering scheme produces a clustering tree, a dendrogram, where at each stage there is one cluster less than at the previous stage. In order to select the number of clusters, we need to evaluate the overall clustering solution at each stage. A solution would be to calculate the data likelihood and compare with the BIC criterion [9]. The generalised likelihood ratio should not be used as such, since model complexities differ throughout the tree and have to be compensated.

The change in the BIC measure is equivalent to the BIC distance between the two clusters that are merged. Thus, for hierarchical clustering methods, the BIC measure may be optimised in a greedy fashion so that we calculate the BIC distance between the two closest clusters before they are merged and stop clustering if the distance is negative [9]. This approach is easily extended to online clustering, where a new utterance is included to the cluster most similar to it if the BIC distance between them is positive, whereas negative BIC distance leads to creating a new cluster [68].

Other possibilities in evaluating the clustering solution include calculating the penalised within-cluster dispersion [35] or estimating the cluster purity [66]. The penalised within-cluster dispersion performed well in a speaker adaptation and may

be recommended for offline speaker clustering. However, when applied to an online clustering task, the measure had a tendency to severely underestimate the number of clusters [47]. The nearest neighbour purity estimator [66] gives a good estimate of the true cluster purity only if there are sufficiently many clusters. Since there are initially no clusters in online clustering, and we do not know how many there will be in the end, the nearest neighbour purity estimator does not necessarily suit this task very well.

Theoretically, online clustering should have a reduced performance compared to the offline methods since it makes local comparisons instead of searching globally for the optimal partition. In speaker turn clustering, however, the results are often better when an online clustering scheme is used [47, 68]. Online clustering probably benefits from the structure of typical speaker segmentation test data like broadcast news audio, where the utterances of one speaker are often close in time [68]. In broadcast news, it is common to have separate news stories that follow one another and have different reporters and interviewees each.

Perfect speaker turn clustering method would not benefit from the broadcast news structure. The difference between offline and online methods performance indicates there may be problems with either the speaker models or with the distance metric. It is possible the problem is in modelling the short speaker turns: in our broadcast news audio, there are many speaker turns shorter than 5 seconds. A model estimated for such speaker turn may be vulnerable to variations not dependent on the speaker, and thus, speaker turns from different speakers may have models too similar. When the speaker turns in broadcast news are labelled with an online method, several speaker turns from the same speaker are encountered before too many speaker turns from other speakers come available, and they are more likely clustered together. A speaker model calculated from several speaker turns probably represents the speaker better than a model calculated from a single speaker turn, and with a better model, some future misclassifications may be prevented.

4.3 Speaker-based adaptation and clustering

We wish to use the speaker segmentation results in speaker-based adaptation, which means speaker-specific transformations will be estimated based on the personalised speaker turns. Speaker adaptation methods are intended to maximise the feature likelihood $P(\mathbf{O}|W)$. Thus, we could consider finding the clustering solution that directly maximises the feature likelihood calculated after speaker adaptation, as proposed in [37]. Instead of calculating distribution models, we would estimate speaker-specific transformations for all the clusters. Each speaker turn would then be assigned to the cluster with the transformation that maximises the feature likelihood in this speaker turn. Note, that we cannot use hierarchical clustering, where each turn is taken as a cluster at first, for the maximum likelihood transformation would always be the one estimated from this speaker turn only. Clustering based on

direct maximisation is best used where the number of clusters does not change, like in k-means clustering.

In k-means clustering, the utterances are divided to k clusters and the clustering solution is updated by moving speaker turns from one cluster to another. To maximise the feature likelihood calculated after speaker adaptation, we would estimate the speaker-specific transformations for all the clusters, and then for each speaker turn we would sequentially try all the transformations and estimate how well the adapted features match the hypothesised transcription of the speaker turn. Speaker turn would then be moved to the cluster corresponding to the winning transformation. After all speaker turns had been found the best cluster, we would recalculate the transformations for the new clusters and repeat reorganising the speaker turns.

Since we have no information on the speakers, we should repeat k-means clustering for several values of k and choose our final solution in much the same way as we would choose the number of clusters in hierarchical clustering. Obviously, this approach is very time consuming, and we conclude that directly maximising the feature likelihood is better suited for an online clustering approach.

Zhang et al. [76] propose an online speaker turn clustering method that is based on maximisation of the feature likelihood calculated after speaker adaptation. They use MLLR, MAP and vector-field smoothing (VFS) for adaptation. Speaker turns are decoded using a speaker-independent model and speaker-dependent models that have been created with speaker adaptation, and the maximum likelihood model is selected. If this is the speaker-independent model, a new adapted model is created based on the speaker turn, and if the selected model is one of the speaker-dependent models, the model is updated with the new data.

In the television and broadcast news audio that we have for speaker segmentation experiments the speaker turns are very short, less then 30 seconds on average, and automatically detected speaker turns are expected to be even shorter, since speaker change detection methods tend to produce spurious speaker change boundaries. Thus, we need a transformation that generalises well even when estimated based on relatively small amount of data. MLLR would be one such transformation. However, the MLLR-adapted models need much memory space, so we should consider finding a transformation that may be applied directly to features [45].

CMLLR is a model-space method, but the CMLLR estimated transformations may be applied to directly to features as described in Section 3.2. In order to estimate the transformations, we need to recognise the input audio and generate a state sequence hypothesis, a time-aligned sequence of acoustic units that are needed to produce the recognised text. The estimated transformations are applied to the features generated with the speaker-independent model, and the same state sequence hypothesis may then be used to evaluate the feature likelihoods that we compare in order to find the best transformation for a given speaker turn. Note that we do not need to re-decode the speaker turns in order to do the comparisons.

Thus, speaker turns are clustered as illustrated in Figure 4.5. Features extracted from the speaker turn are adapted with transformations estimated for previous speakers, if such exist. State information is read from the hypothesis and the log-likelihoods are calculated as [21]

$$L(\mathbf{o}(\tau)) = \ln P(\mathbf{o}(\tau)|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{A}, \mathbf{b}) = \ln P(\hat{\mathbf{o}}(\tau)|\boldsymbol{\mu}, \boldsymbol{\Sigma}) + \frac{1}{2} \ln |\mathbf{A}|^2, \tag{4.6}$$

where $\hat{\mathbf{o}}(\tau)$ are the transformed features and $\mathbf{A}$, $\mathbf{b}$ are the transformation matrix and constant bias. Log-likelihoods are summed over time and thus they become likelihood values for the existing transformations. At speaker change boundary, we then find the highest log-likelihood value. Should this belong to the features calculated with the speaker-independent model, a new speaker cluster is created and the feature information collected from the speaker turn is used to estimate a CMLLR transformation for the new cluster. If instead the maximum likelihood features were produced with a speaker dependent transformation, the speaker turn is added to the corresponding cluster. The feature information is merged with previously collected information (see Section 3.2) and a new transformation is estimated for the cluster. Thus, the proposed method does not only label the speaker turns, but also estimates the speaker-specific CMLLR transformations, and we do not need a separate transformation estimation step as with the speaker model methods.

Often a transformation estimated for one speaker can also improve the likelihood score of another. This would result in both speakers being clustered to the same group. To handle the problem, we add a threshold to accepting the decision made in comparing the likelihoods: if the ratio between the likelihood value calculated for the unadapted features and the highest speaker adapted likelihood does not surpass a given threshold, we take it that the selected transformation would not benefit our current speaker significantly, and we decide we have a new speaker. Such threshold should not be very sensitive to differences in the audio, for speaker adaptation with CMLLR is expected to introduce notable improvements to the likelihood score if the transformation is correct for the speaker.

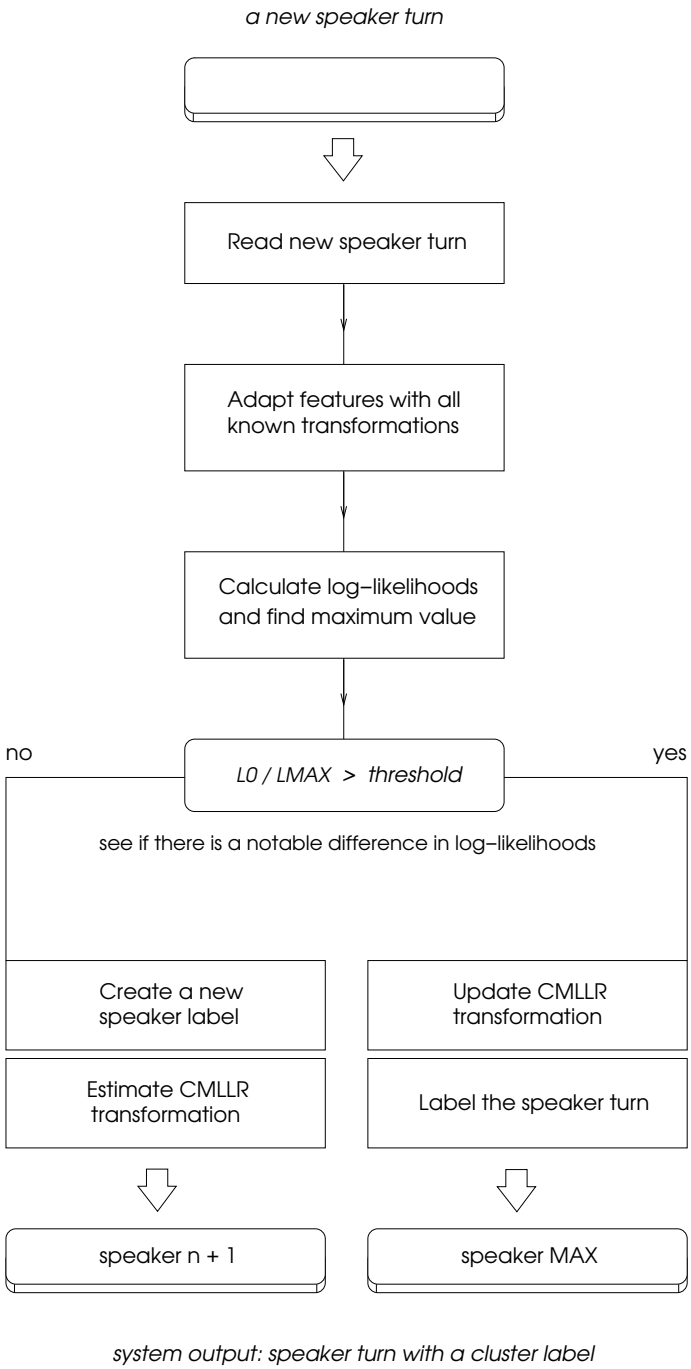
a new speaker turn
Read new speaker turn
Adapt features with all known transformations
Calculate log–likelihoods and find maximum value
no
L0 / LMAX > threshold
yes
see if there is a notable difference in log–likelihoods
Create a new speaker label
Update CMLLR transformation
Estimate CMLLR transformation
Label the speaker turn
speaker n + 1
speaker MAX
system output: speaker turn with a cluster label

Figure 4.5: (Previous page) Our speaker clustering method processes one speaker turn at a time. Information extracted from previous speaker turns is preserved in speaker-specific CMLLR statistics and transformations that correspond to a certain speaker label, and no information on the forthcoming speaker turns is utilised. Speaker turn is adapted with all CMLLR transformations and labelled so that the feature likelihood is maximised. If speaker adaptation can notably increase the feature likelihood, the speaker turn is assigned to the cluster with the transformation corresponding to the maximum likelihood score, and the transformation is then updated with the new data. If speaker adaptation does not increase the feature likelihood much, a new speaker label is created and a new transformation estimated. Likelihood score for the features calculated with the speaker-independent model is denoted with L0, and the maximum likelihood score for the adapted features with LMAX.

Chapter 5

Experiments

We test speaker segmentation methods and speaker-based adaptation in a large vocabulary speech recognitions task with English broadcast news data and Finnish television news audio. News broadcast data is difficult for automatic speech recognition systems, for there are typically several speakers and different environments. Speaker adaptation could improve the recognition results, but it is first needed that we find the speaker change boundaries and label the detected speaker turns. Short speaker turns that are typical for broadcast news audio make the speaker segmentation and adaptation task challenging, but on the other hand, online clustering methods that we use for labelling the speaker turns may benefit from the broadcast news structure.

In this work, we seek to find a speaker segmentation method that could provide the speaker turns for speaker adaptation purposes. Thus, speech recognition results are in a key role when we evaluate the outcome. For all the experiments, we utilise the large vocabulary speech recognition system developed in the Laboratory of Computer and Information Science at the Helsinki University of Technology. The speech recognition system is described in Section 5.1 along with the speaker segmentation methods implemented for this work. The selected broadcast news data is discussed in Section 5.2. Section 5.3 describes the evaluation metrics, and the results are presented in Section 5.4.

5.1 System

Our large vocabulary continuous speech recognition system is much like the system described in Chapter 2. Speech signal is represented with 12 MFCC and the log-energy along with their first and second differentials. Features are calculated in 16 ms windows with 8 ms overlap. Cepstral mean subtraction (CMS) and a maximum likelihood linear transformation that is estimated in training are applied to the

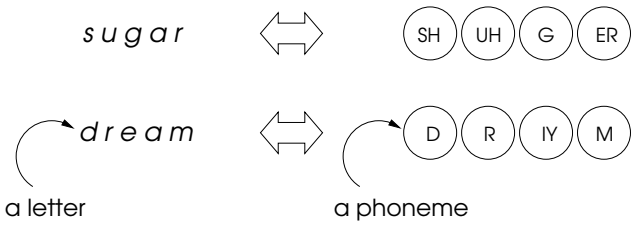

Figure 5.1: CMUDICT format phoneme representations for *sugar* and *dream* [1].

features. For acoustic modelling we have state-clustered Hidden Markov triphone models constructed with a decision-tree method [54]. Our English model has 5062 states modelled with 32 Gaussians. Finnish model has 1860 states that are each modelled with 8 Gaussians. State durations are modelled with gamma probability functions in both English and Finnish [58].

Acoustic models for English have been trained with Fisher telephone conversations data [11]. The selected training data set has 180 hours of conversational speech data from wide variety of speakers. The different pronunciations for American English are well-represented as there are different regional dialects as well as speech from non-native speakers. For Finnish models, we have used data taken from the Finnish SPEECON database [34]. The selected 26-hour data set has clean speech recorded with a close-talk microphone from 208 male and female speakers. Among utterances are words, sentences and free speech.

Our speech recognition system uses a growing n-gram language model [64]. English models are word-based and have been trained with the English newswire corpus Gigaword [28]. The pronunciation dictionary for English contains 60000 most frequent words selected from this corpus. Words are mapped to phonemes with the CMUDICT phoneme set as illustrated in Figure 5.1 [1]. For Finnish, we use morphs as the base recognition unit [32]. Finnish language models have been trained with book and newspaper data from the Finnish text collection [2]. Since all words and word forms can be represented with morphs, we have an unlimited decoding vocabulary for Finnish. Our decoder is an efficient time-synchronous beam-pruned Viterbi token pass system [57].

Methods for speaker-based segmentation were implemented for this work. Speaker change detection and speaker turn clustering are written as separate programs. Speaker change detection reads the audio files and state sequence hypothesis generated with the speech recognition system. It uses the moving windows approach with generalised likelihood ratio (GLR) or Kullback-Leibler (KL) distance (see Section 4.1). GLR distance may be used with a threshold calculated with Bayesian information criterion (BIC) and is then referred to as BIC distance. For speaker turn clustering, we have two alternatives: a speaker model based method that uses the BIC criterion (Section 4.2) and a method that combines speaker turn clustering and adaptation as described in Section 4.3. For simplicity, we will refer to the meth-

ods as BIC–STCL (BIC based speaker turn clustering) and ADA–STCL (adaptation based speaker turn clustering). Both speaker turn clustering methods are online in the sense that they label the given speaker turns sequentially. For speaker-based adaptation, we selected constrained maximum likelihood linear regression (CMLLR) that could be found in our speech recognition system and needed not be implemented for this work.

5.2 Test datasets

Speaker-based segmentation and adaptation will be tested with English broadcast news and Finnish television news audio. Broadcast news are often used for testing speaker segmentation. There are several different speakers and the news material is well-suited for speech recognition since planned speech from the newscasters and reporters should correspond well to the news data that has been used in training the language models. However, there are interviews with conversational speech and parts where music is played. Finnish television news also present foreign language material. Thus, we cannot recognise all material in broadcast news and have to select suitable test datasets.

In our English broadcast news and Finnish television news, the news broadcasts comprise several short news stories that each have a similar structure. The newscasters usually list the main news events when the news begin and lead the program from thereafter. Each news story begins with an introduction from the newscaster. Introductions may be very short, but in Finnish television news, the newscasters also deliver short news that do not include interviews. Longer news stories are given for a reporter often located at the scene. Reporters again introduce the news story and share background information. News stories often have speech extracts with political speeches or interviews with civil authorities and first-hand witnesses. The reporters and interviewees usually appear in one news story alone.

We have selected three test sets from the Voice of America (VOA) news broadcasts. VOA 1 contains mostly planned speech from the newscaster and the reporters, but VOA 2 and VOA 3 have also interviews, where there is hesitation and other such phenomena typical for free speech. Interviews are, however, often related to politics or economy, and they do not represent normal conversational speech. Interviews including conversational language or laughter, for example, were discarded from the test sets. VOA 2 and VOA 3 have also parts where music is played in the background, although most parts with background music were not accepted in the test sets. Music and movie extracts were also not included. More information on the VOA evaluation sets is given in Table 5.1. They are each collected from one news broadcast.

In addition to the evaluation sets, we collected a 51-minute development dataset from two VOA news broadcasts. This set is used to find the optimal LM scale that determines whether the decoder should trust on the acoustic models or language

Table 5.1: Evaluation sets

	VOA 1	VOA 2	VOA 3	YLE
total audio (min)	19	38	41	51
# speakers	10	28	27	41
# speaker turns	41	88	80	128

models more and to set the thresholds in speaker change detection. Development set has planned speech from the reporters and some interviews. It can be segmented to 128 speaker turns and has speech data from 41 different speaker. Some speakers from the development set are also present in the test set.

One evaluation set is taken from the Finnish Broadcasting company (YLE) evening news. The selected audio contains only planned speech from the newscasters and reporters, but may have some music or other noise in the background. Interviews were discarded for the language is quite informal in them. Music and foreign language speech were not included either. Thus, we have mostly news story beginnings with the introductions from the newscaster and the reporter. There is speech data from 7 television news broadcasts and from 49 different speakers as indicated in Table 5.1. We also have a short 10-minute development set for finding the threshold parameters for the YLE television news data. LM scale is optimised using a dataset with Finnish broadcast news and radio talks. They contain planned speech with no background noise.

VOA and YLE datasets are segmented into news stories. We keep this segmentation, so the news story changes are automatically marked as speaker change boundaries, and only the speaker change boundaries within the news stories need to be found with speaker change detection. There are 125 unmarked speaker change boundaries in the VOA evaluation sets and 58 unmarked boundaries in the YLE evaluation set. When we evaluate the speaker change detection performance, we consider the unmarked speaker change boundaries alone. Thus, there are no automatic true detections due to the news story boundaries. Note, however, that when speaker turn clustering and speaker adaptation are tested with automatically detected speaker turns, they will benefit from the news story boundaries that are accurate.

When speaker turns are labelled, the news story boundaries are taken as regular speaker change boundaries, and thus, they do not convey any additional information for speaker turn clustering. Speaker labels are given for one dataset at a time, so a speaker who appears in more than one news stories in the same dataset should be given the same label every time, but if this speaker should appear in another dataset, he or she will receive a new label.

Experimental results suggest that speaker segmentation and adaptation performance decreases when the number of speakers increases [3]. YLE evaluation set has the

least amount of speech data per speaker, and could thus be considered the most difficult evaluation set. However, the YLE evaluation set is all planned speech with 16 kHz sampling rate, whereas the VOA broadcast news are sampled with 8 kHz. With higher frequencies lost, speech recognition results may become less accurate for female speakers.

In order to compare speaker adaptation with the true speaker turns and with the speaker turns detected and labelled using speaker segmentation methods, the evaluation sets were hand-segmented to speaker turns with speaker-specific labels. In the YLE evaluation set, the speaker turns are marked very carefully and do not include the silences in between the speaker turns. The detected speaker change boundaries are considered correct if they are found in the silence period. However, there may not be more than one detected speaker change between any two speaker turns. The silences are very short, and in the VOA evaluation sets, the silences between speaker turns are not even marked. Instead, single speaker change boundaries have been set between the speaker turns.

5.3 Evaluation metrics

Speech recognition performance is commonly measured with word error rate (WER). In Finnish, however, letter error rate (LER) is better suited for the purpose, for words tend to be rather long. Finnish words most often correspond to more than one English word and consist of several concatenated morphemes like the word "kahvin+juoja+lle+kin" which would be "also for a coffee drinker" in English. Word and letter error rates are derived from the Levenshtein distance also known as edit distance [49]. The idea is to compare recogniser output to a reference text and count the dissimilarities: units replaced with another unit (substitution error), units added (insertion error) and units missed (deletion error). Error rate is then calculated as

$$\text{WER / LER} = \frac{S + I + D}{N} \cdot 100\%, \tag{5.1}$$

where S, I, D denote the unit substitution, insertion and deletion errors, respectively, and N is the total unit count in the reference text. In addition to words and letters, phonemes are sometimes used as units.

Speaker adaptation performance is evaluated through changes in word and letter error rates. Error rates are calculated for the news stories, and the average result is reported. However, the average results may be better after speaker adaptation if the changes are notable enough in some news storied even if the results on other news stories would have decreased. In order to state that a method is expected to improve speech recognition results on random news stories, we need to test the differences for statistical significance. We have paired data, for the same news stories are evaluated before and after speaker adaptation. Dependent t-test is the standard test to determine if paired measurements are identical in mean. However, for the t-test to be

valid we need to assume the samples are normally distributed. Wilcoxon signed-rank test is a nonparametric alternative that may be applied when the distribution is not known for certainty [51].

Wilcoxon signed-rank test first calculates the difference in word or letter error rates between the test sets A and B for each test sample separately. In this work, samples correspond to news stories. The differences are then ranked according to the absolute value. The smallest absolute change in the word or letter error rate is given rank 1, the second is given rank 2 and so on. Positive ranks are those that correspond to positive differences: the results are better in the test set B than in the test set A. Negative ranks correspond to negative differences respectively. In two-tailed Wilcoxon test [51], both positive and negative ranks are summed, and the lesser of the two sums is adopted as test measure. The differences between error rates in test sets A and B are regarded as statistically significant if the test measure exceeds certain critical value that depends on the sample size. Critical values used in this work are calculated at significance level $p = 0.05$.

Note that the Wilcoxon signed-rank test compares only the differences in the performance measure and does not consider the performance level. Thus, the difference between LER 15 % and LER 16 % is regarded as equal with the difference between LER 6 % and LER 5 % although the latter improvement is more difficult to achieve. It could be more appropriate to use logarithmic word and letter error rates if a rank test is conducted on the results, for some news stories are much more difficult than others, and the differences in word and letter error rates are quite notable.

When combined with speaker adaptation, speaker segmentation performance reflects to speaker-specific transformation estimates, and thus, to speech recognition results. If the speaker turns are to be used for something else than speaker adaptation, other evaluation metrics may be more suited, for speaker adaptation may sometimes favour a solution that deviates from the correct speaker segmentation result. For example, it may be beneficial to create different speaker labels for speaker and environment combinations rather than just different speakers, assuming that there is sufficiently speech data available for each combination.

To evaluate speaker segmentation results in isolation from speaker adaptation we focus on the speaker turns. Speaker change detection performance is evaluated with the SCD evaluation metrics suggested in [46]. Given the true speaker change intervals $[a_i\ ,\ b_i]$, where $a_i = b_i$ if we have no silence marked in between the speakers, the errors are defined as follows: false acceptance occurs when a hypothesised speaker change boundary does not fall into any interval $[a_i - \alpha\ ,\ b_i + \alpha]$, and false rejection occurs when there are no detected speaker change boundaries within $[a_j - \alpha\ ,\ b_j + \alpha]$, where α is a tolerance factor. False acceptance and rejection errors are illustrated in Figure 5.2.

The results are reported in terms of false acceptance rate (FAR) and false rejection rate (FRR). False acceptance rate determines the percentage of false acceptances among the detected speaker change boundaries, and false rejection rate the percent-

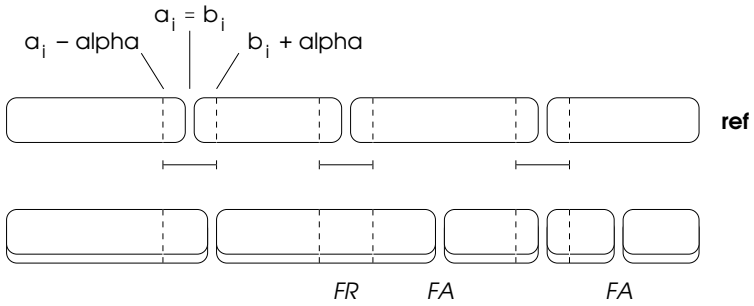

Figure 5.2: Detected speaker change boundaries are regarded as correct if they are within distance α from a true speaker change boundary defined as an interval $[a_i\,,\,b_i]$. False acceptance (FA) errors occur when speaker change boundaries are detected outside the accepted speaker change intervals $[a_i-\alpha\,,\,b_i+\alpha]$, and false rejection (FR) errors when accepted speaker change intervals are passed without any speaker change boundaries being detected.

age of true speaker change boundaries that were not detected. We are more willing to tolerate false acceptances than false rejections in speaker change detection, as false boundaries can be made disappear in speaker turn clustering. However, false boundaries do make the detected speaker turns shorter, and may thus hamper speaker turn clustering performance. False acceptance rate and false rejection rate are inversely related: many true speaker change boundaries are detected when the threshold T is set low, but also many false detections will then be made, and the other way around if the threshold T is set high.

Speaker turn clustering performance may be evaluated with average cluster purity (ACP) and average speaker purity (ASP) introduced in [3]. They may be calculated at utterance or frame level, but when speaker turn clustering is preceded with speaker change detection, the frame-level evaluation is more suited. Cluster purity is then defined as the probability that two frames taken from one cluster at random and with replacement both came from the same speaker [66]. The first frame taken from cluster i is spoken by speaker j with probability $n_{ij}\,/\,c_i$, where n_{ij} denotes the number of frames in cluster i that are spoken by speaker j and c_i the total number of frames in cluster i. Since the first frame is returned to the cluster before the second is drawn, the probability that both frames are spoken by speaker j is $n_{ij}^2\,/\,c_i^2$. Furthermore, the probability that the two frames are spoken by any common speaker is given as

$$p_i = \sum_{j=1}^{S} n_{ij}^2\,/\,c_i^2, \tag{5.2}$$

where S is the number of speakers. Cluster purity $p_i = 1$ if the frames in cluster i are all from the same speaker, and close to 1 if most are from a common speaker. If there is data from many different speakers in the cluster, the purity is smaller. For example, if frames in cluster i would be evenly divided between k speakers, the

cluster purity would be $1 / k$ [66]. Speaker purity for speaker j is similarly given as

$$q_j = \sum_{i=1}^{C} n_{ij}^2 / s_j^2, \tag{5.3}$$

where C is the number of clusters, n_{ij} the number of frames in cluster i that are spoken by speaker j, and s_j the total number of frames spoken by speaker j. Speaker purity is the probability that two frames taken from one speaker at random and with replacement both are in the same cluster and have the same speaker label.

The average cluster purity thus measures to what extent the frames in a random cluster all come from the same speaker, and the average speaker purity similarly measures to what extent the frames from a random speaker are all labelled the same. The average cluster and speaker purity are calculated as the weighted averages [3]

$$\text{ACP} = \frac{1}{C} \sum_{i=1}^{C} p_i \, c_i \tag{5.4}$$

$$\text{ASP} = \frac{1}{S} \sum_{j=i}^{S} q_j \, s_j \, . \tag{5.5}$$

Note that these measures should not be used alone to evaluate a clustering solution, for the average cluster purity would score high if there were as many clusters as there are frames, and on the other hand, the average speaker purity would score high if the frames were all in the same cluster. Neither would be a sensible solution to our problem. There are other measures like Rand Index [33] and BBN Metric [66] that combine the average cluster and speaker purity in the sense that they favour pure and large clusters.

5.4 Results

Speaker segmentation and adaptation experiments are conducted as follows: First the input audio is decoded with the speaker-independent models to obtain a state sequence hypothesis. Speaker change detection is then applied in order to divide the audio to speaker turns. Results from experiments with different distance metrics used in speaker change detection are presented in Section 5.4.1. Speaker turns are clustered, and labelled according to the clustering solution. Speaker model based and speaker adaptation based clustering methods are compared in Section 5.4.2. With personalised speaker turns, speaker-specific transformations may be estimated for speaker adaptation. The input audio is re-decoded after speaker adaptation and the final speech recognition results are presented in Section 5.4.3.

5.4.1 Speaker change detection

Speaker change detection (SCD) aims at locating speaker change boundaries and thus dividing the input audio to speaker turns. Speaker change detection techniques were discussed in Section 4.1. In this work, we will use the moving windows approach. Window size is 700 frames. We use a state sequence hypothesis generated with the speaker-independent model and test for speaker change at every hypothesised phone boundary. A single Gaussian with diagonal covariance is used in modelling the feature distribution inside both windows, and the three distance measures presented in Section 4.1 are tested in calculating the distance between the two windows. Speaker change is found where the distance passes a threshold T which is set experimentally. Note that the Bayesian information criterion (BIC) defined complexity penalty given in Equation 4.3 is calculated for a single full-covariance Gaussian model. For single Gaussians with diagonal covariances, the complexity penalty is $p \log(N + M)$, where p is the feature dimension and N, M are the number of features in the feature sets we wish to compare.

We notice the generalised likelihood ratio (GLR) distance (Equation 4.2) and the BIC distance, GLR distance with the BIC complexity penalty as a threshold, have a similar behaviour if the scaling parameter λ is included in the complexity penalty and tested with different values like the threshold T that we use with the GLR distance. Without the scaling parameter, the BIC distance performance is not comparable to that of GLR distance with a hand-set threshold. Since the two distance measures have near to identical behaviour when the BIC distance has the scaling parameter, we need not test both, but will use the GLR distance alone. The preliminary tests with the different threshold and scaling parameter values were conducted with the VOA development set.

The preliminary tests with different threshold values aim at finding the optimal threshold for the selected distance measure. In testing with the GLR distance, we also noticed that changing the threshold most affects the false acceptance rate and not the number of false rejections. This suggests that the GLR distance finds most speaker changes with high robustness, but on the other hand, is blind to the rest. When we tested Kullback-Leibler (KL) distance with different threshold values, both false acceptances and false rejections were affected. Setting the threshold T so that false acceptance and false rejection errors are equally likely, we would have FAR and FRR around 20 % for both distance metrics. Threshold values are, however, set to favour false acceptances over false rejections as discussed in Section 4.1.

Results from testing GLR distance (Equation 4.2) for speaker change detection are shown in Table 5.2 and results from speaker change detection with the KL distance (Equation 4.5) in Table 5.3. Tolerance factor α (see Section 5.3) is set to 2 seconds for the VOA evaluation sets and to 0.5 seconds for the YLE evaluation set. The tolerance factor is set more tight for the YLE evaluation set because the silences in between the true speaker turns are marked in the reference files (see Section 5.2). Since the silences are typically very short, the results between the VOA and YLE

Table 5.2: Speaker change detection results using moving windows approach and generalised likelihood ratio distance. False acceptance rate (FAR) and false rejection rate (FRR) are calculated with tolerance factor 2 seconds for the VOA evaluation sets and tolerance factor 0.5 seconds for the YLE evaluation set.

	VOA 1	VOA 2	VOA 3	YLE
FAR (%)	26.3	23.9	12.2	51.0
FRR (%)	12.5	12.1	17.3	15.5

Table 5.3: Speaker change detection results using moving windows approach and Kullback-Leibler distance measure. False acceptance rate (FAR) and false rejection rate (FRR) are calculated with tolerance factor 2 seconds for the VOA evaluation sets and tolerance factor 0.5 seconds for the YLE evaluation set.

	VOA 1	VOA 2	VOA 3	YLE
FAR (%)	30.4	32.1	20.7	50.5
FRR (%)	0.0	8.6	11.5	17.2

evaluation sets would be more comparable if the tolerance factor for VOA would be closer to 1 second. The false acceptance rate and the false rejection rate for the VOA evaluation sets with 1 second tolerance factor are, however, very high compared to the YLE results, and do not allow comparing between the GLR and KL distances as comfortably as the results calculated with the tolerance factor in 2 seconds. Note that small deviations in the speaker change boundary locations are not expected to degrade the speaker adaptation performance.

The average false rejection rate calculated over the VOA evaluation set test results is 14 % with the GLR distance metric (see Table 5.2) and 7 % with the KL distance metric (see Table 5.3), and the speaker change detection results are better with the KL distance metric in every VOA evaluation set individually, too. However, speaker change detection with the KL distance also produces more false boundaries than speaker change detection with the GLR distance. The average speaker turn length after KL distance based speaker change detection is 24 seconds. This should be enough for our speaker turn clustering or speaker adaptation methods, and thus, we will use KL distance for speaker change detection in the VOA evaluation sets. For the YLE evaluation set, we will use GLR distance, for it has a better false rejection rate then KL distance. The average speaker turn length is around 18 seconds with both distance metrics, and altogether, the two metrics were quite equal when tested on the Finnish television news audio.

As the detected speaker turns are on average shorter than the true speaker turns, we could benefit from local clustering. This is expected to remove spurious speaker

Table 5.4: Speaker change detection results after local clustering. Speaker turns for the VOA evaluation sets were detected using moving windows approach and Kullback-Leibler distance, and for the YLE evaluation set using moving windows approach and generalised likelihood ratio distance. False acceptance rate (FAR) and false rejection rate (FRR) are calculated with tolerance factor 2 seconds for the VOA evaluation sets and tolerance factor 0.5 seconds for the YLE evaluation set.

	VOA 1	VOA 2	VOA 3	YLE
FAR (%)	6.7	8.6	4.7	43.5
FRR (%)	12.5	8.6	21.2	17.2

change boundaries, and thus, to decrease the the false acceptance rate (see Section 4.1). We use local clustering with the BIC distance metric and model the feature distribution in every speaker turn as a single Gaussian with full covariance matrix (Equation 4.3). Local clustering is tested without a threshold or a scaling parameter, and the results are presented in Table 5.4.

Local clustering removes the false boundaries efficiently from the VOA evaluation sets, and after the procedure, the average false acceptance rate calculated over the test sets is 7 %. Before local clustering we had the average FAR 28 % as indicated in Table 5.3. However, local clustering also deletes some correct speaker change boundaries from VOA 1 and VOA 3 evaluation sets, and the average false rejection rate increases to 14 %. The detected speaker turns are also too few in number. If we set a negative threshold for the BIC distance, we can have the average FRR 8 % and FAR 18 %, but then we also have two threshold parameters to optimise: one for speaker change detection and one for local clustering. Local clustering does not change the results on the YLE evaluation set as much as the VOA results.

5.4.2 Speaker turn clustering

We test speaker turn clustering first with the true speaker change boundaries and then with automatically detected boundaries in order to assess how sensitive our clustering methods are to errors introduced in speaker change detection. We will test BIC criterion and speaker model based clustering method (BIC–STCL) discussed in Section 4.2, and speaker adaptation and likelihood score based method (ADA–STCL) that combines speaker turn clustering and constrained maximum likelihood linear regression (CMLLR) as described in Section 4.3. Both methods are online in the sense that they label speaker turns one at a time based on information extracted from the previous and not the forthcoming speaker turns. A new speaker turn can be given a new speaker label or the label can be selected amongst the most recently used speaker labels. The number of available speaker labels is set to 10 in order to limit the processing time.

Table 5.5: BIC–STCL results when the system is given the true speaker change boundaries. Average cluster purity (ACP) measures how well clusters are limited to one speaker, and average speaker purity (ASP) measures how well speakers are limited to one cluster.

	VOA 1	VOA 2	VOA 3	YLE
ACP	1.00	0.95	0.97	1.00
ASP	0.93	0.93	0.93	0.47

Table 5.6: ADA–STCL results when the system is given the true speaker change boundaries. Average cluster purity (ACP) measures how well clusters are limited to one speaker, and average speaker purity (ASP) measures how well speakers are limited to one cluster.

	VOA 1	VOA 2	VOA 3	YLE
ACP	0.98	0.85	0.88	0.97
ASP	0.97	0.90	0.85	0.96

In BIC–STCL, speaker turns are modelled as single full-covariance Gaussians and the distance or dissimilarity between the models is measured with the BIC distance (Equation 4.3). Results on VOA and YLE evaluation sets are presented in Table 5.5. In this experiment, the true speaker change boundaries were provided for the system. The average cluster purity (ACP) calculated over the VOA evaluation sets is then 0.97 and the average speaker purity (ASP) is 0.93. The results indicate that the speaker turns clustered together usually originate from the same speaker, and on the other hand, the speaker turns from one speaker are usually limited to one cluster. Note that mislabelled speaker turns exist even when ACP is close to 1.00, for there may be more speaker labels than there are speakers. The extra labels are not many in the VOA evaluation sets, but our YLE evaluation set with ASP 0.47 could spare some twenty speaker labels.

ASP 0.50 would mean that some 30 % of the available data is practically discarded when speaker-dependent transformations are estimated. However, in YLE evaluation set, there is an imbalance between the speakers, and the low ASP value is partly due to system assigning more than one label to a newscaster from whom there is more speech data than what we have from the other speakers. The mistake does not necessarily show in the speaker adaptation results, as there should be enough data under both labels to estimate the speaker-dependent transformations properly.

Results from speaker turn clustering with the ADA-STCL method are in Table 5.6. VOA 1 and YLE evaluation sets have been labelled better than with the BIC–STCL method, but the performance on VOA 2 and VOA 3 is substantially inferior to the

Table 5.7: BIC–STCL results when the system is given automatically detected speaker change boundaries. Speaker change boundaries for the VOA evaluation sets were detected using moving windows approach and Kullback-Leibler distance, and for the YLE evaluation set using moving windows approach and generalised likelihood ratio distance. Average cluster purity (ACP) measures how well clusters are limited to one speaker and average speaker purity (ASP) measures how well speakers are limited to one cluster.

	VOA 1	VOA 2	VOA 3	YLE
ACP	1.00	0.91	0.90	0.98
ASP	0.97	0.84	0.89	0.45

Table 5.8: ADA–STCL results when the system is given automatically detected speaker change boundaries. Speaker change boundaries for the VOA evaluation sets were detected using moving windows approach and Kullback-Leibler distance, and for the YLE evaluation set using moving windows approach and generalised likelihood ratio distance. Average cluster purity (ACP) measures how well clusters are limited to one speaker and average speaker purity (ASP) measures how well speakers are limited to one cluster.

	VOA 1	VOA 2	VOA 3	YLE
ACP	0.98	0.84	0.83	0.96
ASP	0.97	0.87	0.82	0.84

previous results. These sets are difficult in the sense that there are sections such as interviews that contain unplanned speech. It is possible those sections contain more recognition errors than CMLLR estimation can tolerate, and the estimated transformations are not close enough to the optimal speaker-dependent transformations that could discriminate between speakers.

The previous tests were conducted with the true speaker change boundaries, but in speech data collected from any real-world situation, the speaker change boundaries are most likely missing. Thus, also the speaker turn clustering methods should be evaluated based on their performance when the true speaker turns are replaced with automatically detected ones. We use moving windows approach and KL distance measure (Equation 4.5) for speaker change detection in the VOA evaluation sets and GLR distance measure (Equation 4.2) for speaker change detection in the YLE evaluation set. Results from speaker turn clustering with the detected speaker turns are presented in Table 5.7 and Table 5.8.

Both ACP and ASP have generally decreased compared to the test with true speaker turns, but this was inevitable given that we are completely missing some speaker

change boundaries. VOA 1 results have not decreased since there were no missing boundaries. BIC–STCL results on VOA 1 are actually better than the results obtained with the true speaker turns, for ASP value has raised from 0.93 to 0.97. It is likely that the spurious speaker change boundaries introduced in speaker change detection have divided some speaker turns in a manner that whatever prevented the system from recognising the true speaker has been segmented out as illustrated in Figure 5.3. Note that the same phenomenon can lead into decrease in ASP as also illustrated in Figure 5.3.

To find out whether errors in speaker turn clustering are due to the detected speaker turns not having enough data, we test if local clustering (see Section 4.1) can improve the results. BIC distance based local clustering with a hand-set negative threshold T as discussed in the previous section does not change the results much from those presented in Table 5.7 and Table 5.8. BIC–STCL results for VOA 3 improved as ASP value increased to 0.89, but there were no other changes, and ADA–STCL results did not change. Thus, it appears the methods are not inefficient because there would not be enough data to estimate the speaker models or speaker-specific CMLLR transformations, but the reason is strictly in the misplaced speaker change boundaries.

Last, we have the ADA–STCL likelihood score threshold (see Section 4.3) to discuss. For the experiments reported here, the threshold was set so that there should be at least 10 % likelihood improvement due to the speaker-dependent transformation in order to have the transformation approved for current speaker. Most often the difference is over 15 % if the transformation is correct and should be approved, and less than 5 % if not. Completely removing the threshold would lead to a huge drop in ACP values. With YLE evaluation set ACP would be under 0.70 and with VOA data under 0.55 in all individual evaluation sets.

5.4.3 Speaker-based adaptation

Speaker segmentation and adaptation are tested under three different conditions: with the true speaker turns and speaker labels, with the true speaker change boundaries but with the speaker labels missing, and with no prior information on speakers. Thus, we first recognise the audio with the speaker-independent model in order to produce a state sequence hypothesis. Then, if speaker change boundaries are not marked, we use speaker change detection to find them, and if speaker labels are missing, we use speaker turn clustering. CMLLR transformations (see Section 3.2) are estimated based on given speaker turns and the audio is re-decoded after speaker adaptation. Thus, speaker adaptation is evaluated based on performance in automatic speech recognition task. Speech recognition performance is evaluated with word error rate (WER) and letter error rate (LER).

Speech recognition and speaker adaptation results on VOA and YLE evaluation sets

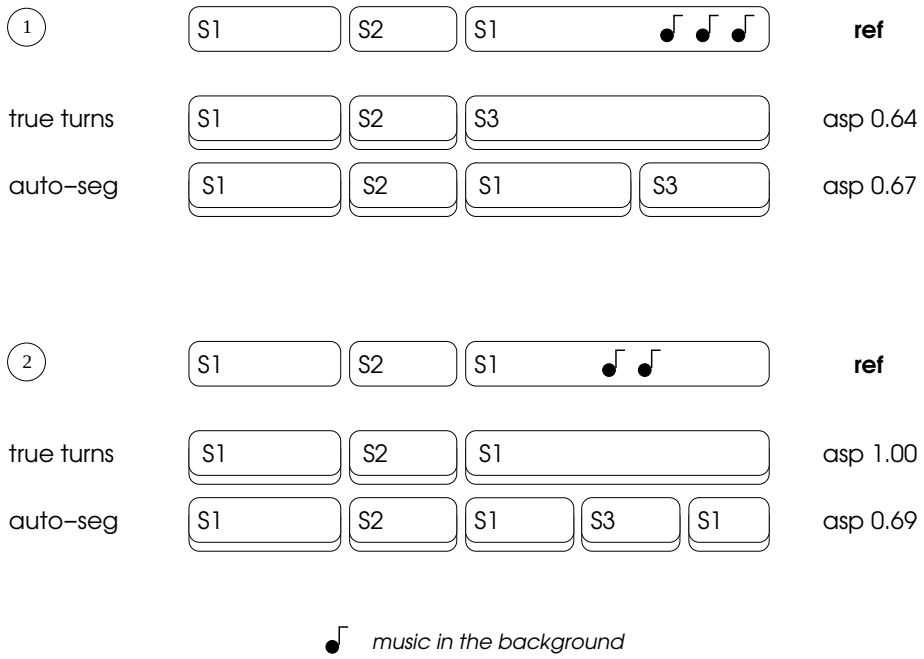

Figure 5.3: Sudden changes in background conditions often create false speaker change boundaries in automatic segmentation. False boundaries are not welcome in general, but in times they might actually improve speaker clustering performance. If there is, for example, background music present in the speaker turn for some time, it is possible that the speaker in this turn is not recognised. If this speaker turn should then be divided into parts with and without background music, speaker clustering would likely label the other part correctly thus increasing average speaker purity as illustrated in the first example. On the other hand, if the speaker turns had been labelled correctly when the true speaker change boundaries were available, false boundaries may instead encourage speaker clustering to errors as illustrated in the second example. Note, however, that although there is a serious decrease in the average speaker purity, excluding the parts with background music may prove beneficial when estimating the speaker-specific transformations for speaker adaptation.

Table 5.9: Speech recognition and speaker adaptation results. We have baseline results with the speaker-independent model and without adaptation (A), and speech recognition results after speaker-based adaptation when the speaker-dependent CMLLR transformations are estimated based on (B) true speaker turns and labels or (C) true speaker turns clustered with BIC–STCL or (D) true speaker turns clustered with ADA–STCL. Furthermore, we have speech recognition results when speaker segmentation and adaptation are applied without any prior information about the speakers i.e. when the transformations are estimated based on speaker turns detected using moving windows approach and Kullback-Leibler distance for the VOA evaluation sets and generalised likelihood ratio distance for the YLE evaluation set and clustered (E) with BIC–STCL or (F) with ADA–STCL.

(a) WER (%)

	VOA 1	VOA 2	VOA 3	YLE
A	25.3	30.3	29.5	23.0
B	22.6	27.3	25.9	19.8
C	22.5	28.5	26.0	19.9
D	22.5	28.3	26.0	19.5
E	22.3	28.2	25.6	20.0
F	22.3	28.3	26.0	19.4

(b) LER (%)

	VOA 1	VOA 2	VOA 3	YLE
A	12.3	15.6	16.2	7.9
B	10.7	13.7	13.8	6.0
C	10.6	14.5	13.8	6.3
D	10.6	14.4	13.8	5.9
E	10.4	14.3	13.6	6.4
F	10.4	14.4	13.9	5.9

are presented in Table 5.9. Baseline results for the VOA evaluation sets range from WER 25 % to WER 30 % and for YLE data we have WER 23 %. English material usually has a lower word error rate than Finnish material because Finnish words can correspond to several English words. VOA news broadcasts having a lower word error rate than YLE television news may be due to a mismatch between training and test conditions, for the acoustic models for English were trained with telephone conversations [11], and the language models are trained with news material [28]. News material would seem an appropriate choice, but in fact, the VOA broadcast news are more relaxed than most conventional news in both topic and style.

Results indicate that CMLLR speaker adaptation significantly improves the speech recognition performance. The relative error reductions in letter error rates are some 13 % on average in the VOA evaluation sets when the speaker-dependent CMLLR transformations are estimated based on true speaker information (Table 5.9 B). In the YLE evaluation set, the relative error reduction is on average 24 %. The results

from speaker adaptation with the true speaker turns and speaker labels are the reference we wish our speaker segmentation and adaptation results to reach.

Results from speaker adaptation when the speaker turns are clustered and labelled automatically (Table 5.9 C and D) are indeed very similar to the previous results. Speech recognition results on VOA 2 are on average not on the same level with the results with the true speaker turns and labels, but the difference is not statistically significant. VOA 3 results have not changed notably, and VOA 1 and YLE results are on average better than with the true speaker turns and labels when the YLE evaluation set is clustered with ADA–STCL, but the difference is again not statistically significant. Statistical significance is tested with the Wilcoxon signed-rank test (see Section 5.3).

It is noteworthy that the speaker adaptation results on the VOA evaluation sets for BIC–STCL clustered speaker turns are not better than the speaker adaptation results for ADA–STCL clustered speaker turns even though the average cluster and speaker purity were much better for BIC–STCL (see Table 5.5 and Table 5.6). This suggests that a speaker segmentation method may be well-suited for speaker adaptation even if it does not label the speaker turns correctly, but rather makes its own decisions on this matter. On the other hand, there is a clear difference between BIC–STCL and ADA–STCL average results on the YLE evaluation set. The difference is not statistically significant according to the Wilcoxon signed-rank test, but nonetheless illustrates how a notable difference in average speaker purities can affect the speaker adaptation results.

When no prior information on speakers is given, results are still much the same. Results (Table 5.9 E and F) are significantly better than the baseline results with no speaker adaptation, but there are no significant differences compared to the results with the true speaker turns and labels or to the previous results. The BIC–STCL results on the VOA evaluation sets and the ADA–STCL results on VOA 1 are on average better than in the previous test with the true speaker turns given (Table 5.9 C and D), and thus, it appears automatic speaker segmentation can, to some extent, even benefit speaker adaptation. However, results on VOA 2 are on average better when the speaker-dependent CMLLR transformations are estimated based on the true speaker turns and labels. The average cluster and speaker purities are about the same for VOA 2 and VOA 3, but VOA 2 contains a little more conversational speech and has more word recognition errors than the other evaluation sets, and speaker adaptation with the true speaker turns and labels also improved the VOA 2 results less than the results on other evaluation sets.

Whether one should generally choose BIC–STCL or ADA–STCL for speaker turn clustering remains an open question. Speaker segmentation and adaptation results on VOA 2 and VOA 3 evaluation sets are on average better when the detected speaker turns are clustered with BIC–STCL, but on the other hand, results on VOA 1 are the same for both speaker turn clustering methods, and ADA–STCL results on the YLE evaluation set are on average better both with the true and detected

Table 5.10: Speech recognition results when the CMLLR transformations used for adaptation are estimated (A) based on news stories, without speaker segmentation and (B) based on detected and labelled speaker turns. Speaker turns are detected using moving windows approach and Kullback-Leibler distance for the VOA evaluation sets and generalised likelihood ratio distance for the YLE evaluation set and labelled with BIC–STCL in the VOA evaluation sets and with ADA–STCL in the YLE evaluation set.

(a) WER (%)

	VOA 1	VOA 2	VOA 3	YLE
A	23.3	29.1	26.6	20.1
B	22.3	28.2	25.6	19.4

(b) LER (%)

	VOA 1	VOA 2	VOA 3	YLE
A	11.0	15.0	14.4	6.3
B	10.4	14.3	13.6	5.9

speaker turns. Speech recognition results for VOA 1 and YLE are more accurate than the results for VOA 2 and VOA 3. The recognition errors probably affect the CMLLR transformation estimation needed in ADA–STCL more when we have the automatically detected speaker turns. Automatically detected speaker turns are on average shorter than the true speaker turns, and thus, the errors in the state sequence hypothesis may not be averaged out as normally happens. Since BIC–STCL does not utilise the state sequence hypothesis, its performance is not directly dependent on the speech recognition result.

All and all, speaker adaptation has introduced some significant error reductions with the true and the automatically generated speaker turns equally well. However, we do not actually know, whether this indicates that the detected speaker turns are good. CMLLR transformations are quite simple and thus not capable in learning the finer speaker characteristics, and it is possible the transformations are equally effective whenever estimated and applied on the same, relatively small, data collection. To determine whether this is the case, we estimate a CMLLR transformation for each news story, and compare the results.

Results from news story based adaptation (Table 5.10) are on average inferior to the speaker adaptation results with either the true or detected speaker turns, and thus, using speaker segmentation would still introduce relative error reductions around 5 % on the VOA evaluation set and some 6 % on the YLE evaluation set. Statistical testing finds the differences in error rates insignificant, which, however, does not mean that speaker segmentation and adaptation do not work, but rather that speaker adaptation does not work as we assumed with this particular data. We expected speaker adaptation would remove speaker-dependent characteristics, but CMLLR also compensates environmental differences. Acoustic models for neither

English nor Finnish were trained with broadcast news data, and thus, the environmental differences may affect speech recognition more than the differences between individual speakers. Results could have been different had we tested speaker segmentation and adaptation on data that matches the acoustic models and language models better and speaker adaptation could focus on truly modelling the speakers. For example, the CSR [24] corpora with news articles read from the Wall Street Journal could suit our models well.

Since there are most often only 1 – 3 different speakers in a news story, news story adaptation may also be too close to speaker adaptation. If the audio had not been divided into news stories, and we would have estimated a common transformation for the whole news broadcast, there would probably be a much greater difference between those results and speaker adaptation results. Also, if we had the speaker turns, but we would not have them clustered but would just estimate a CMLLR transformation for each speaker turn, the speaker adaptation results would probably degrade because there would not be enough data to properly estimate the transformations. In this sense, it is a fair comparison between speaker adaptation and adaptation based on news stories, for there are approximately as many detected speakers as there are news stories when the VOA evaluation sets are clustered with BIC–STCL and the YLE evaluation set with ADA–STCL.

Chapter 6

Conclusions and Discussion

Inspiration for this work came from the desire to apply speaker adaptation in automatic speech recognition of news broadcast data where we have multiple speakers and no prior information on them. We wanted to automatically segment the data to speaker turns that are labelled according to the speaker. We used a common speaker change detection method and compared speaker model based clustering and a new clustering method that seeks to maximise the feature likelihood when features are adapted using constrained maximum likelihood linear regression (CMLLR) estimated for each cluster separately.

We were interested in finding speaker turns that would be accurate enough to guide speaker-based adaptation, and hence speech recognition results were used to evaluate the speaker segmentation performance. However, we wanted to assess the speaker segmentation performance also in isolation from speaker adaptation in order to find any characteristics the chosen methods may have. Also, if one wanted to use speaker segmentation for spoken document content analysis or alike, it would be important that the speaker turns as such are correct.

We tested metric-based speaker change detection with Kullback-Leibler distance and generalised likelihood ratio based distance. Results were quite even between the two metrics. Most speaker change boundaries were found, but as noted in [38], metric-based speaker change detection has a tendency to produce many false boundaries. Local clustering can be used to remove some false boundaries, but this did not affect the speaker turn clustering or speaker adaptation performance in our experiments. In this work, a speaker change is detected if the distance passes a hand-set threshold, but in the future, it would be better to have an adaptive threshold [48].

Both speaker turn clustering methods had average cluster purity and average speaker purity well over 0.8 both with the true and detected speaker turns, and the speech recognition performance after speaker adaptation was the same when the speaker-specific CMLLR transformations were estimated based on the true speaker turns and labels, and when automatic speaker segmentation methods were used to produce the

speaker information. It is mentioned in [3] that average cluster and speaker purities should be greater than 0.70 on average in order to benefit speaker adaptation in applications like broadcast news transcription.

The average speech recognition results on VOA 1 and YLE improved when CMLLR transformations were estimated based on the true speaker turns with automatically set speaker labels instead of the true speaker labels. Similar results were reported in [76]. Furthermore, when speaker change detection was used to generate the speaker turns, we noticed further improvements in the average speech recognition results calculated after speaker adaptation. We observed from the YLE television news audio, that speaker change detection would often set false speaker change boundaries around sections with some short-term noise, and the sections were then given a different label than the other speech data from the same speaker. It seems reasonable, that the recognition results improve, if we estimate separate CMLLR transformations for certain speaker with and without background music, for example. If speaker segmentation results were not intended for speaker adaptation we would not wish to have mislabelled speaker turns, but in our case, a mistake is welcome if it improves speaker adaptation and speech recognition performance.

The speaker turn clustering methods tested in this thesis have one difference. We found that the speaker adaptation based speaker turn clustering method introduced in Section 4.3 achieved much better average cluster and speaker purities with test sets that had better speech recognition results. Thus, speaker model based clustering is probably more suited for speaker turn clustering if one fears the speech recognition results contain many errors. More tests with different data should, however, be conducted to compare the speaker turn clustering methods, since the speaker adaptation based method performed notably better than the speaker model based method when applied on the Finnish television news audio.

Another difference between the YLE and VOA evaluation sets is found when comparing the speaker adaptation and speech recognition results. Speaker adaptation improved the letter error rate 24 % on average when applied on the YLE evaluation set and around 13 % when applied on the VOA evaluation sets. The baseline results without speaker adaptation are also better for the YLE evaluation set, and since the CMLLR transformations are estimated based on the speech recognition results, the recognition errors in VOA could degrade the speaker adaptation performance. However, the acoustic models for English are also much more complex than the acoustic models for Finnish: English models have over 150 000 Gaussians, whereas Finnish models have around 15 000. It is possible that one transformation common to all states may not be an adequate solution for a model so complex, and we should probably divide the state models to regression classes and estimate the speaker-specific transformations for each class independently [42]. This is common with, for example, maximum likelihood linear regression, but not with CMLLR.

After testing with adaptation based on news stories rather than speaker turns, we came to the conclusion that speaker adaptation did not concentrate on learning

speaker characteristics but rather on some general differences between the training and test conditions. Broadcast news feature some different environments and background noises, whereas the training data for the English acoustic models contained only telephone conversations, and the training data for the Finnish acoustic models contained clean speech recorded with a close-talk microphone. CMLLR transformations can also handle environmental differences, so we still do adaptation, but not speaker adaptation. The problem with mismatched environmental conditions is more severe with some other speaker adaptation methods like the eigenvoice approach [52].

If we wanted speaker adaptation to focus on speaker-specific details only, we should probably estimate a CMLLR transformation over several speakers to compensate for the mismatch between the training and testing conditions. The problem with news broadcasts is that in addition to different speakers, there are different environments that would each need a different transformation.

Speaker segmentation has been researched extensively over the years, and it would be difficult to find a rock unturned, a method that is truly new to this field. However, we have implemented a speaker segmentation system that works quite well and enables us to use speaker adaptation in speech recognition tasks with multiple speakers involved, and using CMLLR and the likelihood maximisation scheme in speaker turn clustering proved quite successful. It would be interesting to test the method using models trained with speaker adapted data and see if the results change. Models trained with speaker adapted data do not learn to compensate for speaker variation like normal speaker-independent models.

The problem with our current system is that it cannot separate speech and music. Broadcast news are often punctuated with music, and thus, an automatic broadcast news transcription system would essentially need to distinguish between speech and music sections. Speaker segmentation would then be applied to the speech sections only. Other research areas related to broadcast news transcription task would be language recognition and modelling conversational speech. Language recognition is needed to separate sections spoken in the target language from the sections spoken in other languages that the system cannot recognise, and similar techniques can be applied to dialect detection also. Modelling conversational speech would improve speech recognition performance on interview sections, where incomplete sentences, filled pauses and other such phenomena are common. Last, speech recognition under noisy conditions could improve the results, for sections with background music were found rather common in the YLE television news and in the VOA broadcast news.

Bibliography

[1] *CMU dictionary v. 6.0*. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[2] *Finnish text collection*. http://www.csc.fi/english/research/software/ftc.

[3] J. Ajmera, H. Bourland, I. Lapidot, and I. McCowan. Unknown-multiple speaker clustering using HMM. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 573–576, Denver, Colorado, United States, September 2002.

[4] J. Ajmera, I. McCowan, and H. Bourland. Robust speaker change detection. *IEEE Signal Processing Letters*, 11:649–651, 2004.

[5] T.W. Anderson. *An Introduction to Multivariate Statistical Analysis*. John Wiley & Sons, Inc., 1958.

[6] F. Bimbot, I. Magrin-Chagnolleau, and L. Mathan. Second-order statistical measures for text-independent speaker identification. *Speech Communication*, 17:177–192, 1995.

[7] J. P. Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85:1437–1462, 1997.

[8] L. Canseco-Rodriguez, L. Lamel, and J.L. Gauvain. Speaker diarization from speech transcripts. In *Proceedings of the 8th International Conference on Spoken Language Processing*, pages 601–604, Jeju Island, Korea, October 2004.

[9] S. Chen and P. Gopalakrishnan. Speaker, environment and channel change detection and clustering via the Bayesian information criterion. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, United States, February 1998.

[10] C. Chesta, O. Siohan, and C.H. Lee. Maximum a posteriori linear regression for hidden Markov model adaptation. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, pages 211–214, Budapest, Hungary, September 1999.

[11] C. Cieri, D. Miller, and K. Walker. The Fisher corpus: a resource for the next generations of speech-to-text. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal, May 2004.

[12] T.M Cover and J.A. Thomas. *Elements of information theory.* John Wiley & Sons, Inc., 2nd edition, 2006.

[13] M. Creutz. Adaptering i automatisk taligenkänning (Adaptation in automatic speech recognition). Master's thesis, Helsinki University of Technology, 2000.

[14] M. Creutz and K. Lagus. Unsupervised discovery of morphemes. In *Proceedings of the ACL Workshop on Morphological and Phonological Learning*, pages 21–30, Philadelphia, Pennsylvania, United States, July 2002.

[15] S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-28:357–366, 1980.

[16] P. Delacourt and C.J. Wellekens. DISTBIC: A speaker-based segmentation for audio data indexing. *Speech Communication*, 32:111–126, 2000.

[17] J.R. Deller, Jr., J.H.L. Hansen, and J.G. Proakis. *Discrete-time processing of speech signals.* IEEE Press, 2000.

[18] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[19] V.V. Digilakis, D. Ritchev, and L.G. Neumeyer. Speaker adaptation using constrained estimation of Gaussian mixtures. *IEEE Transactions on Speech and Audio Processing*, 3:357–366, 1995.

[20] E. Eide and H. Gish. A parametric approach to vocal tract length normalization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 346–348, Atlanta, Georgia, United States, May 1996.

[21] M.J.F. Gales. Maximum likelihood linear transformations for HMM–based speech recognition. *Computer Speech and Language*, 12:75–98, 1998.

[22] M.J.F. Gales. Cluster adaptive training of hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 8:417–428, 2000.

[23] M.J.F. Gales and P.C. Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, 10:249–264, 1996.

[24] John Garofalo, David Graff, Doug Paul, and David Pallett. *CSR-I Complete.* Linguistic Data Consortium, 1993.

[25] J.L. Gauvain and C.H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298, 1994.

[26] H. Gish and N. Schmidt. Text-independent speaker identification. *IEEE Signal Processing Magazine*, 11:18–32, 1994.

[27] H. Gish, M. Siu, and R. Rohlicek. Segregation of speakers for speech recognition and speaker identification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 873–876, Toronto, Canada, May 1991.

[28] David Graff. *English Gigaword*. Linguistic Data Consortium, 2003.

[29] A. Gunawardana and W.J. Byrne. Discounted likelihood linear regression for rapid speaker adaptation. *Computer Speech and Language*, 15:15–38, 2001.

[30] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87:1738–1752, 1990.

[31] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2:578–589, 1994.

[32] T. Hirsimäki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkönen. Unlimited vocabulary speech recognition with morph language models applied to Finnish. *Computer Speech and Language*, 20:515–541, 2006.

[33] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[34] D. Iskra, B. Grosskopf, K. Marasek, H. van den Heuvel, F. Diehl, and A. Kiessling. SPEECON – speech databases for consumer devices: Database specification and validation. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 329–333, Las Palmas, Canary Islands, Spain, May 2002.

[35] H. Jin, F. Kubala, and R. Schwartz. Automatic speaker clustering. In *Proceedings of the DARPA Speech Recognition Workshop*, Chantilly, Virginia, United States, February 1997.

[36] Q. Jin, K. Laskowski, T. Schultz, and A. Waibel. Speaker segmentation and clustering in meetings. In *Proceedings of the NIST Rich Transcription 2004 Spring Meeting Recognition Evaluation Workshop*, Montreal, Canada, May 2004.

[37] S.E. Johnson and P.C. Woodland. Speaker clustering using direct maximisation of the MLLR-adapted likelihood. In *Proceedings of the 5th International Conference on Spoken Language Processing*, pages 1775–1778, Sydney, Australia, December 1998.

[38] T. Kemp, M. Schmidt, M. Westphal, and A. Waibel. Strategies for automatic segmentation of audio data. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1423–1426, Istanbul, Turkey, June 2000.

[39] P.K. Kuhl. Auditory perception and the evolution of speech. *Human evolution*, 3:19–43, 1987.

[40] R. Kuhn, J.C. Junqua, P. Nguyen, and N. Niedzielski. Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech and Audio Processing*, 8:695–707, 2000.

[41] M. Kurimo, A. Puurula, E. Arisoy, V. Siivola, T. Hirsimäki, J. Pylkkönen, T. Alumäe, and M. Saraclar. Unlimited vocabulary speech recognition for agglutinative languages. In *Proceedings of the Human Language Technology conference – North American chapter of the Association for Computational Linguistics annual meeting*, City of New York, New York, United States, June 2006.

[42] C.J. Legletter and P.C. Woodland. Flexible speaker adaptation using maximum likelihood linear regression. In *Proceedings of the ARPA Spoken Language Technology Workshop*, pages 104–109, Austin, Texas, United States, January 1995.

[43] C.J. Legletter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9:171–185, 1995.

[44] X. Lei, J. Hamaker, and X. He. Robust feature space adaptation for telephony speech recognition. In *Proceedings of the 9th International Conference on Spoken Language Processing*, pages 773–776, Pittsburgh, Pennsylvania, United States, September 2006.

[45] D. Liu, D. Kiecza, A. Srivastava, and F. Kubala. Online speaker adaptation and tracking for real-time speech recognition. In *Proceedings of the 9th European Conference on Speech Communication and Technology*, pages 281–284, Lisbon, Portugal, September 2005.

[46] D. Liu and F. Kubala. Fast speaker change detection for broadcast news transcription and indexing. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, pages 1031–1034, Budapest, Hungary, September 1999.

[47] D. Liu and F. Kubala. Online speaker clustering. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 572–575, Hong Kong, April 2003.

[48] L. Lu and H. Zhang. Speaker change detection and tracking in real-time news broadcasting analysis. In *Proceedings of the ACM Multimedia*, pages 602–610, Juan-les-Pins, France, December 2002.

[49] I. McCowan, D. Moore, J. Dines, D. Gatica-Perez, M. Flynn, P. Wellner, and H. Bourland. On the use of information retrieval measures for speech recognition. Technical report, IDIAP Research Institute, 2005.

[50] D.T. Merino. Speaker compensation in automatic speech recognition. In J.C. Junqua and G. van Noord, editors, *Robustness in language and speech technology*. Kluwer Academic Publishers, 2001.

[51] J.S. Milton and J.C. Arnold. *Introduction to probability and statistics*. McGraw-Hill, Inc., 3rd edition, 1995.

[52] P. Nguyen, C. Wellekens, and J.C. Junqua. Maximum likelihood eigenspace and MLLR for speech recognition in noisy environments. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, pages 2519–2522, Budapest, Hungary, September 1999.

[53] H. Ning, M. Liu, H. Tang, and T. Huang. A spectral clustering approach to speaker diarization. In *Proceedings of the 9th International Conference on Spoken Language Processing*, pages 2178–2181, Pittsburgh, Pennsylvania, United States, September 2006.

[54] J.J. Odell. *The use of context in large vocabulary speech recognition*. PhD thesis, Cambridge University, 1995.

[55] M. Pitz and H. Ney. Vocal tract normalization equals linear transformation in cepstral space. *IEEE Transactions on Speech and Audio Processing*, 13:930–943, 2005.

[56] D. Pye and P.C. Woodland. Experiments in speaker normalisation and adaptation for large vocabulary speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1047–1050, Munich, Germany, April 1997.

[57] J. Pylkkönen. An efficient one-pass decoder for Finnish large vocabulary continuous speech recognition. In *Proceedings of the 2nd Baltic Conference on Human Language Technologies*, pages 167–172, Tallinn, Estonia, April 2005.

[58] J. Pylkkönen and M. Kurimo. Duration modeling techniques for continuous speech recognition. In *Proceedings of the 8th International Conference on Spoken Language Processing*, pages 385–388, Jeju Island, Korea, October 2004.

[59] L. Rabiner and B.H. Juang. *Fundamentals of speech recognition*. Prentice Hall PTR, 1993.

[60] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[61] H. Robbins. The emprirical Bayes approach to statistical decision problems. *The Annals of Mathematical Statistics*, 35:1–20, 1994.

[62] M.A. Siegler, U. Jain, B. Raj, and R.M. Stern. Automatic segmentation, classification and clustering of broadcast news audio. In *Proceedings of the DARPA Speech Recognition Workshop*, Chantilly, Virginia, United States, February 1997.

[63] V. Siivola. An adaptive method to achieve speaker independence in a speech recognition system. Master's thesis, Helsinki University of Technology, 1999.

[64] V. Siivola and B. Pellom. Growing an n-gram language model. In *Proceedings of the 9th European Conference on Speech Communication and Technology*, pages 1309–1312, Lisbon, Portugal, September 2005.

[65] K.C. Sim and M.J.F. Gales. Adaptation of precision matrix models on large vocabulary continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 97–100, Philadelphia, Pennsylvania, United States, March 2005.

[66] A. Solomoff, A. Mielke, M. Schmidt, and H. Gish. Clustering speakers by their voices. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 757–760, Seattle, Washington, United States, May 1998.

[67] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataraman. MLLR transforms as features is speaker recognition. In *Proceedings of the 9th European Conference on Speech Communication and Technology*, pages 2425–2428, Lisbon, Portugal, September 2005.

[68] A. Tritschler and R. Gopinath. Improved speaker segmentation and segments clustering using the Bayesian information criterion. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, pages 679–682, Budapest, Hungary, September 1999.

[69] L.F. Uebel and P.C. Woodland. An investigation into vocal tract length normalisation. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, pages 2519–2522, Budapest, Hungary, September 1999.

[70] A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.

[71] C.E. Williams and K.N. Stevens. Speech and emotions: some acoustical correlates. *The Journal of the Acoustical Society of America*, 52:1238–1250, 1972.

[72] P.C. Woodland. Speaker adaptation for continuous density HMMs: A review. In *Proceedings of the ISCA Tutorial and Research Workshop on Adaptation Methods for Speech Recognition*, pages 11–19, Sophia Antipolis, France, August 2001.

[73] J. Zdansky. BINSEG: An efficient speaker-based segmentation technique. In *Proceedings of the 9th International Conference on Spoken Language Processing*, pages 2182–2185, Pittsburgh, Pennsylvania, United States, September 2006.

[74] P. Zhan and M. Westphal. Speaker normalization based on frequency warping. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1039–1042, Munich, Germany, April 1997.

[75] P. Zhan, M. Westphal, M. Finke, and A. Waibel. Speaker normalization and speaker adaptation – a combination for conversational speech recognition. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2087–2090, Rhodes, Greece, September 1997.

[76] Z. Zhang, S. Furui, and K. Ohtsuki. On-line incremental speaker adaptation with automatic speaker change detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 961–964, Istanbul, Turkey, June 2000.