Phone Duration Modeling Techniques in Continuous Speech Recognition

Master's thesis

Janne Pylkkönen

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory of Computer and Information Science
Otaniemi 2004

Preface

This work was done in the Laboratory of Computer and Information Science of Helsinki University of Technology during years 2003 and 2004. I thank professor Timo Honkela for supervising my work. It would have been impossible to make this thesis without the enormous prior work done in the speech group of the laboratory. I therefore thank my instructor docent Mikko Kurimo for the possibility to work in the research group and also for the valuable comments and corrections he gave me about this theses. I am grateful to Vesa Siivola for suggesting me this particular topic. Him, Panu Somervuo and Teemu Hirsimäki I also thank for the time they have spent discussing with me about everything that has puzzled me. Their ideas and suggestions have had an important role in making this thesis. I finally thank my family, my friends and especially Tiina Lahtinen, for their support and motivation.

Janne Pylkkönen
Otaniemi, April 29, 2004

TEKNILLINEN KORKEAKOULU **DIPLOMITYÖN TIIVISTELMÄ**

Tekijä: Janne Pylkkönen
Työn nimi: Phone Duration Modeling Techniques in Continuous Speech Recognition
Suomenkielinen nimi: Äänteiden kestomallinnustekniikoita jatkuvan puheen puheentunnistusjärjestelmässä

Päivämäärä: 29. huhtikuuta 2004 | **Sivumäärä:** 66

Osasto: Tietotekniikka
Professuuri: Informaatiotekniikka

Valvoja: Prof. Timo Honkela | **Ohjaaja:** Dosentti Mikko Kurimo

Äänteiden kestoilla on tärkeä merkitys puheen ymmärtämisessä. Esimerkiksi suomen kielessä on paljon sanapareja, jotka erotetaan toisistaan lähinnä juuri äänteiden kestojen perusteella. Tällaisten erojen havaitseminen on siten hyvin tärkeää myös automaattisessa puheentunnistuksessa. Kuitenkin nykyiset puheentunnistusjärjestelmät perustuvat usein kätkettyihin Markov-malleihin, joiden kyky mallintaa äänteiden kestoja on melko heikko mallien oletuksista johtuen. Tässä työssä tutkittiin, kuinka näitä malleja voidaan laajentaa ottamaan paremmin äänteiden kestot huomioon ja parantaa siten puheentunnistimien tarkkuutta.

Tässä työssä tutkittiin kolmea erilaista tekniikkaa, joilla Markov-malleihin voidaan sisällyttää paremmat äänteiden kestojen mallit. Teoreettisen tarkastelun lisäksi nämä kolme tapaa myös toteutettiin osaksi Informaatiotekniikan laboratoriossa kehitettyä jatkuvan puheen puheentunnistusjärjestelmää, josta tässä työssä on esittely. Puheentunnistimen avulla tehtiin tunnistustestejä eri tekniikoita käyttäen ja arvioitiin siten niiden keskinäistä paremmuutta. Parhaalla menetelmällä saavutettiin noin 8% suhteellinen parannus tunnistuksen kirjainvirheeseen. Tämä osoittaa, että äänteiden kestojen paremmalla mallinnuksella voidaan saavuttaa etuja automaattisessa puheentunnistuksessa.

Tunnistustestit tehtiin suomenkielisellä aineistolla käyttäen puhujariippuvaa mallia, jolloin puhujasta riippuvat äänteiden kestojen erot saatiin minimoitua. Tämä oli tarpeellista, sillä äänteiden kestoihin vaikuttavat useat tekijät, joista tämän työn yhteydessä otettiin huomioon vain foneemien kontekstit. Myös muiden tekijöiden huomioon ottaminen olisi luultavasti tärkeää yleisemmissä käyttökohteissa.

Avainsanat: puheentunnistus, äänteen kesto, kestomallinnus, kätketyt semi-Markov -mallit, puheen nopeus

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author: Janne Pylkkönen
Title: Phone Duration Modeling Techniques in Continuous Speech Recognition
Title in Finnish: Äänteiden kestomallinnustekniikoita jatkuvan puheen puheentunnistusjärjestelmässä

Date: April 29, 2004 | **Pages:** 66

Department: Computer Science and Engineering
Professorship: Computer and Information Science

Supervisor: Prof. Timo Honkela | **Instructor:** Docent Mikko Kurimo

The duration of phones play a significant part in the comprehension of speech. Finnish, for example, has several word pairs which can be distinguishable mainly by the duration of their phones. In automatic speech recognition, it is very important to detect these differences. Modern speech recognition systems, however, use hidden Markov models, which are deficient in modeling phone durations due to their intrinsic model assumptions. This thesis studied how the acoustic models of a speech recognition system could be improved to handle phone durations more effectively and improve speech recognition accuracy.

Three different techniques for including improved phone duration models in Markov models were studied. The thesis includes a theoretical study of the techniques. The techniques were also implemented in the speech recognition system developed at the Laboratory of Computer and Information Science. An overview of the system is included in the thesis. Using the speech recognition system experiments were carried out to compare the usefulness of the techniques. The best technique achieved about 8% relative improvement in the letter error rate, which proves that improved modeling of phone durations can benefit automatic speech recognition.

Speech recognition experiments were carried out on Finnish material, using speaker dependent models. This guaranteed that speaker dependent variations in phone durations were minimized, which was necessary given that various factors affect the actual duration of phones. This work only accounted for the effect of the phoneme context. In more general applications, inclusion of other factors are probably also necessary.

Keywords: speech recognition, phone duration, duration modeling, hidden semi-Markov models, speaking rate

Contents

Symbols and abbreviations

$S = \{s_j\}$	Set of states in an HMM
$A = \{a_{ij}\}$	Transition probability matrix of an HMM
$B = \{b_j(\boldsymbol{o})\}$	Set of emission probability density functions in an HMM
$\pi = \{\pi_j\}$	The initial state distribution of an HMM
$D = \{d_j\}$	Set of duration probability density functions in an HSMM
$\boldsymbol{O} = \boldsymbol{o}_1, \ldots, \boldsymbol{o}_t$	Observation sequence of acoustic vectors
$Q = q_1, \ldots, q_t$	HMM state sequence
$P(X)$	Probability of X
$p(X)$	Likelihood of X

DFT	Discrete Fourier Transform
EM	Expectation-Maximization
ESHMM	Expanded State Hidden Markov Model
FFT	Fast Fourier Transform
GPD	Generalized Probabilistic Descent
HMM	Hidden Markov model
HSMM	Hidden Semi Markov Model
LER	Letter Error Rate
LPC	Linear Predictive Coefficients
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum A Posteriori
MFCC	Mel-Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLLT	Maximum Likelihood Linear Transformation
MLP	Multi-Layer Perceptron
MMI	Maximum Mutual Information
PDF	Probability Density Function
SNR	Signal-to-Noise Ratio
WER	Word Error Rate

Chapter 1

Introduction

1.1 Speech and speech recognition

Speech is the single most versatile method of communication. From an informative public announcement to an empathic face-to-face conversation, it adapts itself to a diverse field of situations. Whatever the circumstances, the purpose is to deliver a message so that the receivers understand it. In a technical point of view this is a very challenging task. It requires not only the method for transmitting the information, but also a common language and proper signaling. With speech even this is not enough, as apart from the words it is often very important to have also other clues about the context to realize the full meaning of the message. The situation, the mood of the speaker and speaker's relation to the message, to name a few, all alter the way the message can be understood. Taking all this into account humans tend to be surprisingly good in this task of understanding the messages behind the speech. Achieving similar success with automatic speech recognition using computers is nowhere near.

Noting that speech is about transmitting ideas and observations, it is no wonder that computers, which generally lack the ability to process conceptual information, have hard time dealing with it. We can teach a computer something about the language, the words, and the grammar used in the speech, but we are simply unable to get it to understand the message. And to add further difficulty, speech, or the message behind it, can be much more than just the plain words uttered, as mentioned above. Even for us humans it is more difficult to listen an unknown conversation as an outside observer. But if we know the context or some background information about the speakers, we are able to recover surprisingly much about the information in the speech. The automatic speech recognition systems, however, have usually very little knowledge about these. To get the most out of these systems we would hope to get

as much information as possible from the speech alone to compensate for this lack of prior information about the underlying message. And even in this field we still have much to improve.

1.2 Basics of a modern speech recognition system

Automatic speech recognition has been a research area of its own for quite some time and lots of methods have been established over the years. The methods of statistical pattern recognition have been applied for several decades, and the basic structure of the recognizers have remained surprisingly constant since the 1980's (see e.g. [31]). Naturally there have been continuing development in all areas of speech recognition, not only because of the ever increasing computational power available. But even though models get more complex and more difficult recognition tasks are addressed, the underlying structure is still the same.

Figure 1.1 shows simplified blocks of a modern speech recognition system. Speech recognition begins with preprocessing the signal, which usually involves traditional signal processing, like noise suppression and emphasis filtering. Next, like in any pattern recognition system, feature vectors are extracted from the signal. These acoustic feature vectors are then matched to the trained models of acoustics and language in the process of decoding, which finally produces the recognized text.

Hidden Markov models (HMM) [31] have been found to be the most useful way of modeling the acoustics of the speech for recognition purposes. HMMs link together

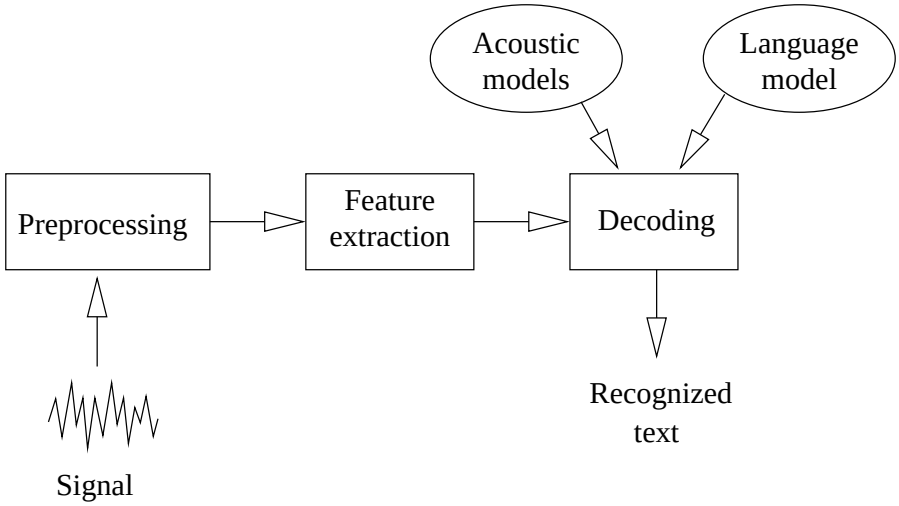


Figure 1.1: The basic functional blocks of a speech recognition system.

the natural time dependence of the speech and the statistical modeling for acoustic features. As a mathematical paradigm HMMs can and have been applied to various areas apart from speech recognition. These include, for example, applications in bioinformatics, recognition of hand written characters (HCR), computer vision and signal processing.

Apart from modeling the acoustics of the speech with HMMs it is evident that in all but the simplest recognizers the language of the speech has to be modeled in some way. In case of a simple connected word recognizer it is enough to list the words and their pronunciations which are expected to be encountered, but if the goal is to recognize continuous speech without restricted vocabulary, things get more complex. This latter task is often referred to as large vocabulary continuous speech recognition (LVCSR). A successful method for modeling the language in such a task collects statistics about the co-occurrences of sequential words and uses this information to predict the probabilities between the different acoustically fitting word alternatives (for construction of such a model, see e.g. [19]). Although this sounds like a very crude way of modeling a language, it is enough for a working speech recognizer, and especially simple enough to be implemented efficiently.

1.3 Goals for this study

As speech recognition is already a well established area of research, there is an existing de facto standard for the recognizers. Certain methods have been proven to be useful, others have been left aside. But as long as there is hope to improve the results of the recognizers, new methods should be evaluated and old, abandoned ones reconsidered in the current context. The purpose of this study is to examine the use of phone durations as an information source to improve an existing speech recognizer. This means introducing explicit duration models for elementary speech units (phones), and implementing these models as a part of the acoustic models of the recognizer.

The phone duration modeling has been studied quite extensively in the past, but the results have not been commonly applied to the modern speech recognizers. This may be because the methods, the actual results and the contexts of the past studies have varied a lot, so there has not been one single solution for the problem. Also the fact that phone durations have no discriminative role in English might have diminished the enthusiasm for studying and applying the methods. On the other hand, Finnish is an example of a language in which phone durations are crucial for the proper comprehension of speech[1]. It is therefore very desirable to find good

[1] As an example, consider the following six Finnish words, which differ only on the durations of their phones: *taka* (back-), *takaa* (from behind), *takka* (fireplace), *takkaa* (an inflection of *takka*), *taakka* (load), *taakkaa* (an inflection of *taakka*). The double letters represent lengthened versions of the phones.

methods for modeling these durations. In this thesis different duration modeling techniques are studied and a unifying comparison of their effect in an LVCSR task is carried out. An emphasis is put into meaningful recognition tests, which would characterize the methods more than traditionally reported error percentages over the test set. The language modeling, although crucial in LVCSR tasks, is left without deeper investigation in this thesis.

The thesis begins with an introduction to acoustic modeling of speech in Chapter 2. Basic acoustic features and the use of HMMs are described. Chapter 3 introduces the speech recognizer used for this study, its specialties and architectural issues concerning the duration modeling. Chapter 4 then describes different methods for modeling phone durations, their pros and cons, along with general considerations for using durations as an information source. A number of tests for the methods have been carried out and they are reported in Chapter 5. Finally, Chapter 6 concludes the thesis, summing up the results and discussing possible future improvements.

1.4 History of speech recognition at our laboratory

Speech recognition has had its place in the Laboratory of Computer and Information Science of Helsinki University of Technology for over two decades. In the early past, some unconventional methods from the present-day view were studied. These include the tests with a speech recognizer utilizing a subspace classifier [15] and the development of a phonetic typewriter based on neural networks [20]. The latter was further extended to include hidden Markov models [42].

In the 1990's more modern-like speech recognition systems were developed. Kurimo studied using self-organizing maps (SOM) and learning vector quantization (LVQ) to train phoneme based speech recognizer utilizing continuous density HMMs [21]. A true large vocabulary continuous speech recognizer for Finnish was then developed in the early 2000's [39], which is the current system and platform also for this study. Recently an important part of the research has been the language modeling of Finnish [40, 7, 39].

Chapter 2

Acoustic Modeling

2.1 Acoustic features

Speech recognizers are basically specialized statistical pattern recognizers. As in any statistical modeling, a good selection of model features is crucial for the performance of the system. For acoustic features to be useful in speech recognition, they should have a number of properties: they should be as descriptive as possible, but at the same time they should not contain excessive redundancy. Moreover, the speech recognizers are not interested in all the information the speech data contains. If the goal is to extract the words behind the speech, all kinds of information about the speaker perceived from his or her voice is unnecessary, as well as the tone or emphasis of the speech. In fact, the less information about these unnecessary qualities the acoustic features contain, the easier it is to model the variations of speech we are actually interested in (here the underlying words) and the better recognizers we are able to build.

As a signal, speech has many distinctive properties which enable us to restrict the analysis and concentrate on the most interesting characteristics it contains. Some of these are (according to [28]):

- **Frequency scale.** The most relevant information in the speech signal lies approximately in the range 200 - 5600 Hz, the same range which human ear is most sensitive to. There is little energy beyond 7 kHz, although humans can hear frequencies up to 20 kHz.
- **Time structure.** Speech signal is quasi-stationary, meaning its characteristics are often similar in time windows of about 20 ms, but remain rarely same for longer than 40 ms.

- **Spectral structure.** Speech segments can be assigned to number of acoustically similar groups according to their spectral properties. For example, the vowels are recognized by large amplitude and relatively stationary signal and discriminated according to their first two or three formant frequencies (resonances of the acoustic tube, giving rise to spectral peaks).

These characteristics originate in the structure and inherent limitations of human speech organs. They have been studied extensively and decent mathematical models exist for them. As a recognition point of view, however, more interesting is how these structures and limitations are perceived in the form of sound, not the way the sound has actually been produced (although studying the physiology of speech organs may help in building relevant models even for recognition purposes). This would suggest that the hearing system should be viewed as a goal in retrieving acoustic information, since there is clear evidence of mutual evolving of speech communication and hearing, like the most relevant frequency range. This is quite intuitive, as it would be strange if speech contained important discriminative information of the form which we couldn't hear. Although this kind of view has also been disputed, the acoustic features most commonly used in the speech recognition field do make use of properties of hearing relevant to this issue [4].

The acoustic feature acquisition is now discussed in detail, partitioned in three sections as in [29], corresponding to the sequential operations performed to the signal.

Spectral shaping

After the speech signal has been converted to digital form with analog to digital (A/D) converter, several signal processing techniques can be applied to it. If, for example, the acoustic or electronic environment has deteriorated the signal, it might be possible to compensate for those in digital domain using normal signal processing methods. However, these are application dependent issues and are not in the scope of this thesis.

The usual case in the speech recognition, at least in the research field, is that the digital speech signal is of good quality and has high enough signal-to-noise ratio (SNR). Then, although there is no need for recovering the signal, it is usual to apply a digital filter before spectral analysis for emphasizing purposes. Traditionally, a very simple first order finite impulse response (FIR) filter is used [29], with a transform function of form

$$H_{\text{emp}}(z) = 1 + a_{\text{emp}} z^{-1}. \tag{2.1}$$

A typical range for a_{emp} is [-1.0, -0.4], and a value of -0.95 is often used. This filter is known as a pre-emphasis filter, and it boosts the high frequencies with 20 dB per decade. The motivation for this kind of processing is a physiological one. On the

one hand, the hearing is most sensitive on the frequencies above 1 kHz, the sensitive range reaching up to about 5 kHz. On the other hand, this same frequency range contains important speech cues, which are significantly lower in amplitude than the low frequency speech components. The effect of the pre-emphasis filter is therefore to equalize the spectrum of the speech signal. The frequencies above 5 kHz are also emphasized, although they do not have that much discriminative meaning. But as those frequencies in speech signal contain very little energy, they do not cause problem in recognition. In addition, the sampling rate usually limits the highest possible frequency to be low enough so that this inapplicable region is not too wide.

Spectral analysis

The speech signal as a time domain waveform contains huge amounts of redundancy. For example, we are able to understand speech even if all the amplitude information is removed by converting speech signal in to a binary waveform [28]. On the other hand, the time structure mentioned in the beginning of this chapter suggests that a good rate for acoustic features would be much less than a typical sampling rate of a speech signal.

For discriminative purposes, the speech is best described in spectral form. At the same time it is possible to remove much of the redundancy, again using the properties of hearing as a guideline. There are several methods for extracting the spectral information from the waveform like using digital filter banks, Fast Fourier Transform (FFT) and Linear Prediction Coefficients (LPC) [29]. These can be further transformed into so called cepstrum coefficients by taking a logarithm from the spectral magnitudes, and then computing the inverse Fourier transform. The cepstrum coefficients resulting from using FFT are nowadays the most commonly used form of spectral information in speech recognition.

When computing the FFT, the spectral resolution, that is, the number of measurement points and their partition, has to be decided. This is not a straightforward task, as a frequency division to equal uniformly spaced bins is by no means optimal. Number of careful studies about the hearing has showed that the frequency resolution of an ear decreases non-linearly as a function of frequency. In accordance with these studies, it would be beneficial to divide the spectrum to frequency bins correlating to the resolution of hearing. A popular approach to this is to transform the frequencies to so called mel scale [29]:

$$m = 2595 \log_{10}(1 + f/700). \tag{2.2}$$

This scale attempts to be perceptually linear, so it can be divided into equal bins. Normally, at most 20 of these bins are used to describe the whole frequency range of a speech signal, which is in accordance to the critical bands of hearing.

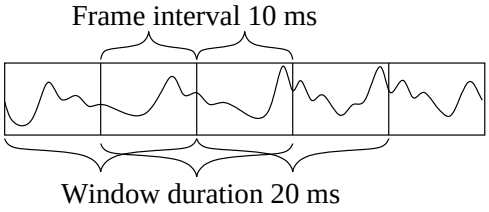


Figure 2.1: Window duration and frame interval.

When extracting spectral information from continuous waveform, there is always the choice between the time and frequency resolution. The longer the time window over which the spectrum is computed, the better frequency resolution is obtained on the cost of the time resolution, and vice versa. The length of the time window is called the window duration. Another question is the rate at which the acoustic features are extracted. This is determined by the frame interval, which should be optimized to the rate at which the spectral information changes, so that to minimize the redundancy of consecutive feature vectors but still maintaining enough time information. According to the properties of speech, the window duration is usually set to about 20 ms and the frame interval to 10 ms. This means that there are overlapping time windows of 20 ms in length starting at every 10 ms. Figure 2.1 illustrates these concepts.

As the spectral magnitude is computed from windowed time segments, the windowing itself alters the spectrum through the convolution effect [30]. To minimize this, the window is a smooth curve weighting the center of the window more than the edges. A usual choice is the Hamming window, which is a special case of the Hanning window. The Hanning window of size N_s is defined as (from [29])

$$w(n) = \frac{\alpha_w - (1 - \alpha_w)\cos(2\pi n/(N_s - 1))}{\beta_w}, \tag{2.3}$$

where β_w is a normalizing constant to get the root mean square value of the window to unity, defined as

$$\beta_w = \sqrt{\frac{1}{N_s}\sum_{n=0}^{N_s-1} w^2(n)}. \tag{2.4}$$

For the Hamming window, $\alpha_w = 0.54$. Figure 2.2 shows Hanning windows with different parameter values.

To sum up, let us formulate the spectral analysis section mathematically. The procedure begins with the Fourier transform of the windowed signal. It is easier to present it in the form of Discrete Fourier Transform (DFT) than with the FFT actually used, but the result is still the same under the constraint that the frequencies are sampled

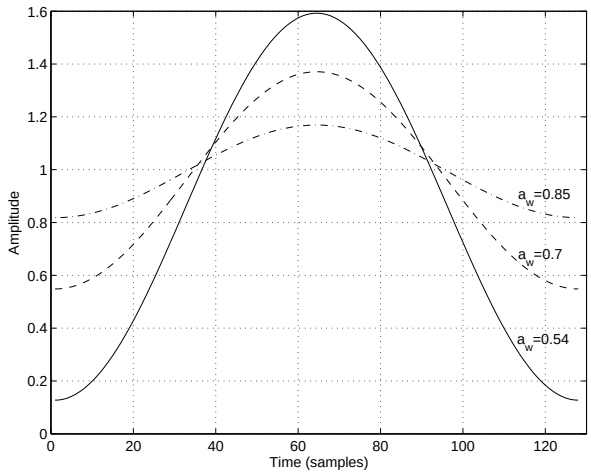


Figure 2.2: Hanning windows.

uniformly [29]. The DFT of a windowed set of samples is defined as

$$S(f) = \sum_{n=0}^{N_s-1} w(n)s(n)e^{-j(2\pi f/f_s)n}, \tag{2.5}$$

where f is the frequency at which the DFT is computed, w(n) is the window from Eq. 2.3, N_s is the window size, f_s is the sampling rate and $s(n)$ is the signal itself. Due to the FFT the spectral magnitudes have to be computed at uniform frequencies. To achieve the mel scale spectrum, the original spectrum is at first oversampled, and these are then averaged with another, frequency domain window. This can be formulated as

$$S_{\text{mel}}(m) = \sum_{n=0}^{N(m)} w_f(m,n)S(f(m,n)), \tag{2.6}$$

where $N(m)$ is the (mel-)frequency dependent size of the window, $w_f(m,n)$ is the window itself, and $f(m,n)$ is a function giving the desired (linear) frequencies for windowing to achieve the approximated mel scale. For the window, overlapping triangular windows are a usual choice, normalized so that each window sums to one.

What is now left is the extraction of the cepstral coefficients by taking a logarithm from the mel spectrum and computing the inverse Fourier transform. As we are interested only on the magnitudes of the cepstrum, not of the phase, we can ignore the imaginary part and write the inverse transform in terms of the cosine basis [30]. The resulting coefficients are known as Mel-Frequency Cepstral Coefficients (MFCC), and they are defined as

$$c(n) = \frac{1}{M_s} \sum_{m=0}^{M_s-1} \log|S_{\text{mel}}(m)| \cos\left(\frac{2\pi}{N_s} mn\right), \tag{2.7}$$

where M_s is the number of mel bins from which the cepstrum coefficients are computed. n belongs to the range $[0, M_s - 1]$, of which all values are not necessarily used as features.

Apart from the spectral magnitudes in the form of MFCC, it is common to use the absolute power of the speech signal as one feature. Actually, the first cepstral coefficient $c(0)$ represents that value, but as there are easier ways to estimate the power, it is not widely used. The actual process of estimating the power is rather straightforward, remembering the time domain windowing it can be presented as

$$P = \sum_{n=0}^{N_s-1} (w(n)s(n))^2. \tag{2.8}$$

Parametric transform

After the signal power and spectrum measurements have been done, these can be considered as the input for the next processing stage. The most common operation performed at this point is the differentiation of the spectral features. They are used to help the further stage of processing (namely the HMMs) to better take the dynamics of the speech spectrum into account. Actually, a lot of information is embodied in the derivatives of the spectral measurements. This follows from the fact that the time aspect is naturally very essential to the speech. The spectrum of the speech stays rarely stationary for more than some tens of milliseconds, but with most pairs of sounds it is gradually changing from one sound to another [28].

As an example, consider the word *fail*, pronounced as [feɪl]. We hear the two vowels in the middle, but as they together form a diphthong, there are actually no two stationary phones but rather a slide from one vowel to another. This kind of pronunciation is much better modeled with trajectories of the spectral components than with the absolute measurements. The trajectories can be modeled by using derivatives of the spectral measurements, or delta features as they are commonly called in discrete cases. One point which also promotes the use of delta features is that when compared to the absolute measurements, they have been noted to be more invariant to the variations of speech caused by the changes in speaker's mental and physical condition [29], therefore constituting more robust and useful features.

Delta features are in a way redundant information, as a good statistical model which is aware of the past features could deduce similar information implicitly. But their properties are so appealing that including them along the absolute spectral measurements is one of the de facto standards in the speech recognition field. In fact, some speech recognizers even include the second order derivatives, or delta-deltas, to the feature vectors to get even more dynamic information to the statistical models explicitly. But it is worth remembering that differentiation is a noisy process [29].

By emphasizing the high frequencies, it tends to amplify the noise in the speech signal. To avoid this, the deltas are usually computed as a smoothed value over several spectral coefficients:

$$\Delta c(n) = \sum_{m=-N_d}^{m=N_d} mc(n+m). \tag{2.9}$$

Window lengths ($2N_d + 1$ in the above equation) of 5 to 9 are common.

The ultimate goal for feature extraction is to get such a measures from the source signal which are easy to be modeled with some statistical model. One property which generally helps the modeling is the independence of the elements of the feature vectors. That is, it is desirable that the measurements (at each time index) are not correlated with each other. From this perspective the cepstral coefficients get more merit than for example direct FFT measures, as they can be regarded to be approximately uncorrelated [29].

This uncorrelatedness is, however, a global property. When the features are grouped to their respective groups, there can be significant correlation among the feature elements. Still it would be beneficial from the modeling point of view to ignore these correlations, as the number of parameters needed for the model can then be vastly reduced. It is not possible to remove all the class-wise correlations and reduce the number of model parameters at the same time, but one can minimize the correlations for example in a maximum likelihood manner by transforming the features before the modeling stage. One rather inexpensive method is to form a single linear transformation, called Maximum Likelihood Linear Transformation (MLLT). This method and its extensions are discussed in [11].

2.2 Hidden Markov models

It is a huge leap to proceed from the acoustic features to the final recognized text. It is obvious that applying statistical pattern recognition to individual features is insufficient to produce natural text, because the time aspect of the signal is so important. That is, it is necessary to segment the sequential acoustic features to groups which would represent basic recognition units, and to also classify these groups of features to be of one specific unit, for example, a certain phoneme. This is not an easy task as both the group region and label are unknown. But at the moment, we have a set of acoustic features which should contain both the time and the acoustic information to solve these unknowns.

Even if the segmentation was given, it is not that straightforward to label the speech segments. A popular method formerly used for labeling the segments is the so called Dynamic Time Warping. It tries to match the given segment to templates, which

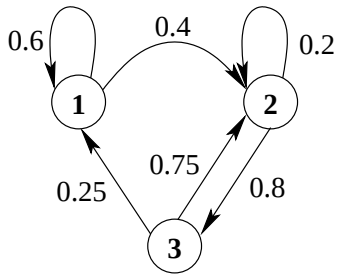


Figure 2.3: A simple Markov chain.

are formed in the training phase. The matching is done using dynamic optimization, by connecting the given speech segment and the template in an optimal, non-linear way. The problem with this method is the high computational cost for comparing the speech segment with a large set of templates. Hidden Markov models (HMM), on the other hand, do not use templates of speech segments but parameterized models to lower the computational cost of labeling a given segment. Furthermore, an HMM can be seen as a very flexible framework, which enables all kinds of modifications to the basic idea.

States and transition

Hidden Markov models are based on Markov chains, a general stochastic paradigm for simulating and describing stochastic processes, developed by a Russian mathematician Andrei Markov in the early 1900's originally for linguistic purposes [24]. What is good with these Markov chains (equally called as Markov processes or Markov models), is that they can be applied to a wide range of stochastic processes. In this context, only discrete Markov chains are considered, although also continuous interpretations exist.

Figure 2.3 shows an example of a simple Markov chain. It consists of discrete states and transitions, or arcs, between the states. A probability is assigned for each transition so that the probabilities of transitions leaving from one state sum to one. This way it is easy to determine the probabilities of state sequences: taking a product of each transition traversed for a certain state sequence produces a correct probability distribution over all the sequences of same length.

The key point in Markov chains is the so called Markov assumption, which is evident in the state presentation of the chain. According to it, the probability of the next state does not depend on the past state sequence but only on the present state. Furthermore, this probability is time invariant. If the set of states in the Markov chain is $\{s_i\}$ and the actual state at time t is q_t, we can formulate these assumptions

as

$$P(q_{t+1} = s_k \,|\, q_t = s_i, \ldots, q_1 = s_j) = P(q_2 = s_k \,|\, q_1 = s_i) = a_{ik}. \tag{2.10}$$

These assumptions give the Markov chain many nice properties and make it easy to handle mathematically. Although restrictive, the chain still remains flexible enough to be able to model a wide range of processes. Besides, it should be noted that any finite discrete process with stationary transition probability distribution conditional only to a finite number of previous states (that is, possibly more than one) can be presented as a Markov chain by introducing auxiliary states to hold the additional information about the past.

Emission probabilities

If we simulate a Markov chain and print out the states which the chain arrives to after each transition, we say that we observe the process. Markov chains become hidden if we can not observe the states themselves but rather samples generated by probabilistic emission functions attached to the states of the chain. A hidden Markov model can therefore be seen as a probabilistic function of the underlying Markov chain, the function being conditional to the state of the chain.

The probabilistic emission functions can be in principle any probability distributions. For one thing, they can be either discrete or continuous. Because in speech recognition the emission distributions are used to model the acoustic feature vectors, it is quite clear that continuous ones are preferred. In the past, however, vector quantization [23, 31] was used extensively to gain some efficiency by using discrete distributions, but as they are no longer considered necessary, they are not discussed in here. Nowadays, the single most widely used emission probability density model is the Gaussian mixture model. It is simply a linear combination of Gaussian density functions, combined so that the result is a valid probability density function. The emission probability density $b_j(\boldsymbol{o})$ for state j is then

$$b_j(\boldsymbol{o}) = \sum_{m=1}^{M_j} c_{jm} \frac{1}{(2\pi)^{d/2}\sqrt{|\boldsymbol{U}_{jm}|}} \exp\left(-\frac{1}{2}(\boldsymbol{o} - \boldsymbol{\mu}_{jm})^T \boldsymbol{U}_{jm}^{-1}(\boldsymbol{o} - \boldsymbol{\mu}_{jm})\right), \tag{2.11}$$

where $\boldsymbol{o}$ is the sample (vector) being modeled, M_j is the number of mixture components, d is the dimension of the vector $\boldsymbol{o}$ and $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{U}_{jm}$ are the mean vector and covariance matrix, respectively, of the mth mixture component of the jth state. The normalization of the probability density is achieved by requiring the mixture coefficients c_{jm} to satisfy the constraints

$$\sum_{m=1}^{M} c_{jm} = 1, \tag{2.12}$$

$$0 \leq c_{jm} \leq 1. \tag{2.13}$$

The covariance matrices $\boldsymbol{U}_{jm}$ are very often constricted to be diagonal, or in some other restricted form, to reduce the huge number of free parameters in high dimensional models.

We are now ready to formalize the structure of an HMM. A hidden Markov model can be described as a tuple $\{S, A, B, \pi\}$, where $S = \{s_j\}$ is the set of states in the HMM, $A = \{a_{ij}\}$ is the probabilistic transition matrix, $B = \{b_j(\boldsymbol{o})\}$ is the set of emission probability density functions and $\pi = \{\pi_j\}$ is the initial state distribution. Next we discuss how all this relates to the actual use of HMMs with speech recognition.

HMMs and acoustic models

The acoustic features can be regarded to be generated by some unknown underlying stochastic process. If we model that process with an HMM, we are able to perform the pattern recognition at the heart of speech recognition. Rather than building one huge HMM to describe all the observations, we model each basic speech unit, let it be a phone or a word, with its own HMM and allow the concatenation of these small HMMs to address the problem at the higher level. As this concatenation is fairly easy by introducing states for entering and leaving the HMM, we now consider only these simple HMMs and requirements for them.

We have already restricted the HMMs used in speech recognition to have their emission probabilities in the form of Gaussian mixture models. Also the structure of the underlying Markov chain is very often restricted to better account for the properties of a speech signal. This is because we are modeling a signal which changes over time, and this progress of time should be incorporated to the model itself. With HMMs, this leads to so called left-right models [31], which have the property that the state index can only increase as time increases. For transition matrix this means the property

$$a_{ij} = 0, \quad \text{if } j < i. \tag{2.14}$$

Often this is even further constrained by allowing transitions only to a few following states. Figure 2.4 shows a common type of HMM which can be used to model, for example, a single phone. Each state have only transitions to itself and to the next state. Last state without transitions does not actually belong to the same HMM, but represents the next HMM to which this HMM has been concatenated to. The model consists therefore of three emitting states, which have been found to be a good configuration for representing the progress of time in a phone, and is therefore considered as the "standard" phone model.

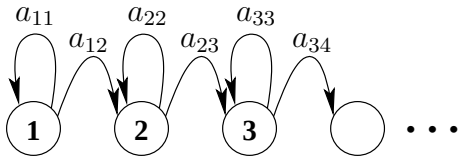


Figure 2.4: An example of an HMM for modeling a phone.

The transition matrix of this HMM is now

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that the last row of the matrix is included just for completeness. To start the Markov chain always from the first state, the initial state distribution is usually set to $\pi = \{1, 0, 0\}$. If the emission probabilities b_j are described as in Equation 2.11 with proper number of mixture components for each state, what is then left is the estimation of the free parameters a_{ij}, c_{jm}, $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{U}_{jm}$. Probably the main reason for the assumptions and choices made to select these models is that an efficient parameter estimation is possible for these kinds of models using a general procedure known as the Expectation-Maximization (EM).

Both using the HMMs for the recognition of speech and re-estimating its parameters with training samples require computing the probability of observing a given set of samples. This corresponds to finding the best path, or equivalently, best taken transitions, over the HMM. Two standard algorithms exist for this problem, the Viterbi algorithm, which finds the single best path over which the probability can be computed, and the Baum-Welch (also known as Forward-Backward) algorithm, which sums the probabilities of all the possible paths together. For more information about the parameter estimation and algorithms used with HMMs, see the excellent tutorial by Rabiner [31].

2.3 The drawbacks of the HMM based acoustic models

Although the methods of speech recognition are established, they leave plenty of room for improvements. Popular spectral feature vectors are not as robust as would be desirable. HMM is just one mathematical model with its assumptions and limitations. Together they form undoubtly useful acoustic models for speech recognition, but they are by no means the final answer. Next, a few shortcomings of the described acoustic models are discussed, along with some proposed alleviations.

Likelihood-based training

The efficient EM-algorithm for training the acoustic models aims at maximizing the likelihood of the training data, given the models. If we denote the concatenation of elementary HMM models (according to the training data) by M, the model parameters by Θ and the acoustic training data by X, the problem of training is then to find the model parameters according to

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, p(X \mid M, \Theta). \tag{2.15}$$

This is an approximation of a more correct form of maximizing the probability of the models given the training data. By using Bayes' rule this can be expressed as

$$P(M \mid X, \Theta) = \frac{p(X \mid M, \Theta) P(M \mid \Theta)}{p(X \mid \Theta)}. \tag{2.16}$$

That is, the likelihood-based training ignores the denominator and the term $P(M \mid \Theta)$, which can be seen as a language model term, independent of the acoustic data [5]. The denominator $p(X \mid \Theta)$ is the likelihood of the training data given the model class and HMM parameters, but not the correct concatenation of elementary HMM models with respect to the data. It is therefore a link between the correct HMM model sequence M and all the incorrect state sequences, which could possibly generate the acoustic training data.

In most cases, the language model term can be neglected, but the implications of leaving the denominator $p(X \mid \Theta)$ out are more severe. When maximizing the plain likelihood, as there is no penalty from the term $p(X \mid \Theta)$, the likelihood of incorrect models may also be improved along with the likelihood of the correct one. This leads to poor discrimination between the models [4], although discrimination is the ultimate goal in the actual recognition task. To relieve the situation, several discriminative training algorithms have been developed, but they tend to suffer from an excessive computational burden. Some of these include the heuristic corrective training [1] and Generalized Probabilistic Descent (GPD) [18]. In [5] a brief introduction is given about Maximum Mutual Information (MMI), Maximum A Posteriori (MAP) and GPD schemes. In that report, also a completely different approach involving neural networks is presented.

Temporal phenomena

The main benefit of using HMMs over other methods of speech modeling is that it is very efficient and effective in modeling the time distortions in the speech signal. But the ability of HMMs to model the various temporal phenomena of speech is actually not that good. The basic assumption of modeling the speech as a piecewise

stationary sequence of short-term feature vectors already ignores the inherent correlation between the sequential acoustic vectors observed on the same HMM state. By definition, they are modeled as a stationary stochastic process, which carry no information about the time structure. This is clearly wrong in cases where the spectral information of the sound is constantly changing (e.g. diphthongs).

The HMM is at its best when modeling slightly larger time spans. A typical duration of distinct sounds is about 80 ms [28], so handling that is indeed crucial. But the human short-term memory of auditory periphery spans over 200 ms [4], about the size of a syllable, and that already the HMM has hard time modeling. And still, even larger scale temporal phenomena do exist, like those of intonation and stress.

As mentioned before, some aspects of speech are better modeled with trajectories than absolute spectral measurements, and this is why the dynamic or delta features are used in conjunction with the absolute measurements in practically all the current speech recognition systems utilizing this framework. This also tries to correct the uncorrelatedness of the sequential feature vectors, as the delta features are computed over several spectral vectors. As a disadvantage, the use of dynamic features can make systems more sensitive to the speaking rate [4].

The longer-term contextual information can be somewhat taken into account by making the elementary HMMs modeling the phones context dependent. These triphone-units improve the recognition of speech so much that they are now part of the de facto standard of the modern speech recognizers. As a consequence, however, the number of different elementary models increases dramatically, as well as the required amount of training material. That is why only a portion of contexts is actually modeled, and for the rest context independent phones, or reduced context diphones are used. What comes to the word and sentence level temporal phenomena, there are no established methods for modeling them.

Markov assumption

The existence of efficient algorithms often implies heavy assumptions, and HMMs are no exception. One can then hope that the real world phenomena do not violate the assumptions too badly for the model to be useful. Judging from the successful applications of HMMs with speech recognition implies that they really constitute a working model, but it is still good to be aware of the conflicts between the model and the real world.

The fundamental assumption with HMMs is that there is an underlying Markov chain, whose transition probabilities satisfy the Equation 2.10. This states that the speech signal is modeled as a quasi-stationary process, which changes its state with probabilities which are only dependent on the present state. As mentioned in the

previous section, the acoustic feature vectors of the stationary part are assumed to be conditionally uncorrelated, but this already is violated in several ways: The feature vectors are computed from overlapping time windows, therefore generating correlation between them, the delta features span even further in time, and the speech itself also contains low level time structure which adds correlation to the sequential feature vectors. The Markov assumption about the conditional transition probability is also a simplification without rigorous justifications. For example, all longer-term phenomena violate this by introducing correlation across several HMM states. Examples of this are the before mentioned intonation and the effects of coarticulation [28], although the use of triphones diminishes the impact of the latter.

These drawbacks emerge from the fundamental properties of HMMs, and are therefore very difficult to alleviate if it is desirable to remain in the traditional HMM framework. For some considerations of alternatives, see the interesting article by Bourlard *et al.* [4].

Duration modeling

As an introduction to the actual topic of this thesis, let us consider one more implication of the underlying Markov chain to the modeling ability of HMMs. Consider an HMM state which has a self transition and transitions to other HMM states. It was assumed that the transition probabilities are time invariant, so we can denote the probability of the self transition with a_{ii}. Then the probability of changing the state from state i at any time instant is $1 - a_{ii}$. From this we can deduce that the probability distribution of durations spent in one HMM state is of form

$$P(d) = a_{ii}^{d-1}(1 - a_{ii}), \tag{2.17}$$

that is, the geometric distribution with parameter a_{ii}.

The form of this distribution is exponentially decreasing. Described with one parameter, the distribution can effectively depict only the mean duration. Beyond that it is unable to model any variations in the duration distributions. This is a severe limitation when considering speech recognition applications. The state durations of HMMs correlate with the phone durations of speech, and many studies show that there is a lot of structure in the duration distributions of phones (see e.g. [9]).

Replacing the inherent geometric distribution with a more general one breaks down the assumption of time invariant transition probabilities. This complicates significantly the training and decoding algorithms optimized for standard HMMs. Nevertheless, a number of methods exist for modeling the phone durations more accurately. The duration distribution can be replaced by some more general parametric distribution, like the gamma [22] or Gaussian distribution [14], or with a less parameterized alternative, like with Markov models [9].

Chapter 3

The Speech Recognition System

3.1 Overview of the system

To be able to evaluate new methods for speech recognition, a complete modifiable speech recognition system is required. For this study, a system developed at the Laboratory of Computer and Information Science was utilized. In this chapter, the system is presented in order to form the context for this study. For further details, refer to [39, 13].

The utilized system is a large vocabulary continuous speech recognizer, designed for research purposes. The structure of the system has been kept highly modular so that it is easy to be modified. On the other hand, it is not the most compact or the fastest implementation, as the clarity of the system has been kept as the main design principle. Majority of the tests with the system have been conducted with Finnish material, and especially the language modeling part has been designed to be suitable in that manner.

The acoustic modeling of the system follows the principles depicted in the previous chapter. The system is easily configurable to use different features, parameter values and models. For this thesis, the configuration was as follows. The speech signal was sampled at the sample rate of 16 kHz. After a pre-emphasis filter, the speech waveform was divided to 16 ms windows, with 8 ms frame interval. As acoustic features, 12 MFCCs and a power, along with the delta features of these were used, thus creating 26 dimensional feature vectors. The feature vectors were modeled with Gaussian mixture models using diagonal covariance matrices, aided by an uncorrelating linear transformation (MLLT).

The context dependent phones (triphones) were modeled with three-state left-right

hidden Markov models with no skip states. Each HMM state had its own Gaussian mixture model with four Gaussian components as its emission density function. For triphone models, the number of triphones was empirically adjusted to the available data. The selection criteria was simply to pick those triphones which had enough data in the training material for training a sufficiently general model. Otherwise a diphone (a phone with only a one-side context) or a monophone model was used. Diphones had to be trained also because the system can not model contexts over language model units, so at each word border a diphone had to be created.

The specialty of the system lies in the language modeling. Instead of words, morphs are used as the language model units. These morphs are morpheme-like units which are discovered in an unsupervised manner from a large corpus [7, 39]. Selecting such a unit instead of words is very important in Finnish, which has huge number of inflectional forms due to extensive use of suffixes and compound words. Using morphs enables covering the vocabulary of the language with magnitudes of fewer language model units as would be necessary with words. A 65000 morph vocabulary was used, for which a normal trigram language model was trained.

Using morphs as language model units has unfortunately a couple of drawbacks. The first is that word breaks are no longer recognized automatically, and thus the recognizer must hypothesize a word break after each morph, and let the language model decide whether that word break is needed. This and the fact that morphs are inevitably shorter than words reduces the context which the trigram language model "sees", as it only considers three sequential morphs of which one or two may be word breaks. On the other hand, word order in Finnish is quite relaxed, so traditional word based n-gram language model might not work that well either. As an additional difficulty, using morphs and triphones couples the selection of language model units and phoneme contexts. As the recognizer has no cross-token (be it a word or a morph) contexts, using morphs means that with each morph boundary we have to use diphones instead of triphones. With short morphs, this may introduce a substantial number of reduced phoneme contexts.

3.2 The decoder

A speech recognition system can be seen to consist of two functional parts. The first part deals with the training of the acoustic and language models, which is done prior to the recognition using large amounts of speech and text data. The actual speech recognizer, the decoder, utilizes these trained models to convert an unknown speech signal to recognized text. As most of the system modifications encountered in this study involve the decoder, it is worth examining a bit deeper what it does and how it has been implemented. A more detailed study of the decoder used for this thesis can be found from [13].

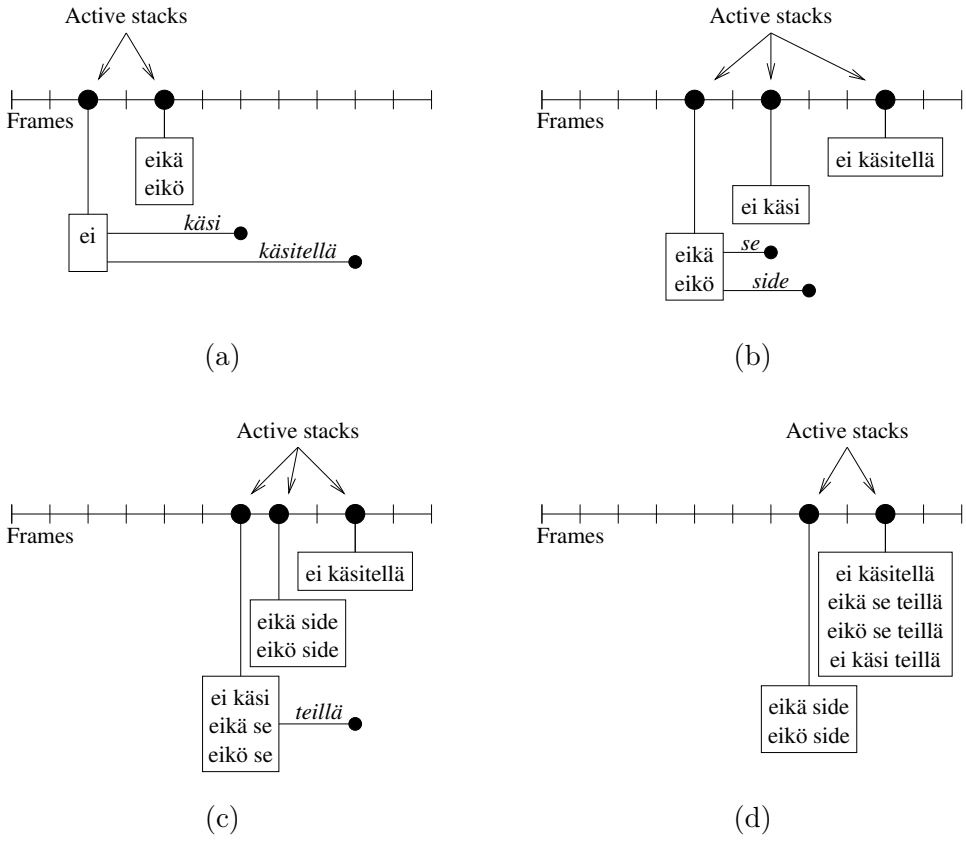


Figure 3.1: A simplified example of the decoding process, proceeding from (a) to (d). Stacks store the hypotheses ending at certain time frame. At each phase, the earliest stack gets expanded. Morph alternatives for expansion are shown in italics. As the stack is expanded, these alternatives are concatenated to the existing hypotheses, and results are inserted to the proper stacks. The order of the hypotheses in the same stack represents their relative likelihood values.

The decoder of the utilized speech recognition system is based on the principle of stack decoding [48]. The recognition proceeds in a time synchronous manner so that local Viterbi searches in a windows of 1.2 seconds are performed from time instances which possibly begin a new morph. The time window corresponds to the approximate maximum duration of the longest morphs. The morph alternatives obtained from the Viterbi searches are concatenated to the hypotheses of the recognized text obtained so far. This results in new hypotheses, which are stored to stacks associated with time frames, according to the ending time of each hypothesis. Several hypotheses may be in the same stack corresponding to one time frame, and each time frame to which one or more hypotheses have ended has its own stack. These stacks are one after another expanded to new hypotheses with the local Viterbi searches, thus proceeding in time. Figure 3.1 illustrates the process.

Using the stacks to store the hypotheses ending at different time instances leads to having many hypotheses stored in parallel. Each hypothesis carry along the likelihood of the match to the acoustic and language models. As the recognition proceeds the worst hypotheses get pruned out, leaving finally only the best matches for the recognition result. This pruning is defined by allowing maximum of 10 hypotheses to be stored in a single stack. This parameter, as well as many others in the decoder, has been empirically adjusted to produce good recognition results in reasonable time.

Pruning at the hypothesis level is not so important as the pruning at the local Viterbi search. This latter is controlled with a so-called beam parameter, and it is the most important factor for defining the tradeoff between the accuracy and efficiency of the decoder. The beam parameter defines how much the likelihoods of the new morph alternatives are allowed to deviate from the best one. In effect, it defines the breadth of the Viterbi search and affects the number of morph alternatives which are used to expand the existing hypotheses. The beam parameter was used extensively in the tests to control the running time of the decoder.

The decoder uses the acoustic data (including phone durations) and language model data as separate knowledge sources, which it combines to one likelihood value for each hypothesis. The combining is done by scaling the log likelihood values of the different knowledge sources and then summing them together. The scaling is necessary because of the numerous assumptions and simplifications incorporated with the models and algorithms used along the recognition process. Due to these inconsistencies it is necessary to give less weight to the acoustic information and more to the language model in order to achieve the best recognition results. It is enough to scale all but one of the log likelihoods of the different knowledge sources. Therefore the acoustic log likelihood is left unscaled, and all the other knowledge sources, namely the transition probabilities, duration model probabilities and language model probabilities, have their own scaling factors. During the experiments, all these scaling factors were optimized with a development set independent of the actual test set to get the maximum recognition accuracy.

Interestingly, the actual scaling of the acoustic likelihood (not the log likelihood) has theoretically no effect to the recognition result when pruning is disregarded. For practical reasons the acoustic likelihoods are, however, normalized so that for each acoustic frame the sum of the likelihoods of all Gaussian mixtures is one. This way the likelihood values are kept within numerical bounds, and it also helps controlling the pruning level of the decoder with the beam parameter.

Chapter 4

Duration Modeling for HMMs

4.1 Phonetic consideration

Variations in phone durations are one form of speech prosody, along with intonation and speaking rate. Their effect may not be observable in the spectral structure, but more in the relationships between the acoustic segments. Even though prosody can vary a lot without affecting identity of the words [28], they are very essential to natural speech, and can even account for the correct understanding. This occurs clearly with phone durations on those languages which have different phoneme lengths. This means that there can be otherwise similar words which differ only on the length of a phoneme, pronounced with different phone durations. But even though there would not be different lengths for phonemes, phone durations still vary on other reasons, like to emphasize syllables or to rhythm the speech. This kind of information can be utilized for recognition purposes.

Need for duration modeling is very apparent in Finnish. There exists several word pairs for which the only distinction is the duration of a phone, and this same is evident also in the written forms of the words. Some examples of these words are *kisa* and *kissa*, *asia* and *aasia*, and *muta* and *muuta*, in which the doubled letter is pronounced as a lengthened version of the original phone. The duration information is used in English too to distinguish between words, for example in words *seat* and *sit*. But in English, the different durations are accompanied with a difference in the acoustic quality of the phones, the quality being the more important cue for discrimination [46]. That is why it can be seen that the words contain phonologically different phonemes, not the same phoneme with different lengths.

As an example, let's consider a comparison between Finnish and English vowels, as carried out by Wiik [45]. He noted that any one of the eight Finnish vowels may

occur as a single or double (as their written and pronounced form), and the distinction between their durations is clear. In Wiik's measurements between the Finnish vowels in primary-stressed contexts, when comparing all the single and double vowels, the double vowels were on average 2.3 times longer in duration than the single ones. In English there are no single and double forms of the vowels, but one can still categorize the vowel phonemes, or vocoids, to short and long ones, based on the pronunciations. Still, for this kind of grouping, similar observations as in Finnish about a distinction between their mean durations could not be made. Only when the observations were made in the same environments (contexts) and with the phonemes with similar phonetic qualities, the categories became clearly distinct. Again with primary-stressed vowels, Wiik measured that the English long vocoids were then on average 1.8 times longer than the short ones in their contexts.

We can now deduce that for some languages the duration modeling can be crucial, as the only difference between the utterances of certain words may be in the durations of their phones. But even with languages where phone durations do not give discriminative information between the words, there may be measurable, deterministic variations in the phone durations. This suggests that analyzing them may give some useful information, for example, about the contexts of the phonemes. This kind of information is always useful and can help the recognition task.

For phone durations to be useful as an information source in speech recognition it would be desirable for them to contain only moderate variation. Unfortunately this is not exactly the case. In fact, several factors affect the duration of a phone [47]: the context, the position of the phoneme in a syllable and the position of the syllable in a word, the number of syllables in a word, the stress, the desire to emphasize the word, and of course the general speaking rate. Also the background for the communication affects the durations: read and conversational speech can have significant durational differences [28]. Some of the durational variances can be taken into account, like the context and the overall speaking rate, but the rest of the variation simply has to be tolerated. In this thesis, only the context is somewhat taken into account by using context dependent phonemes (triphones). The analysis of material is done in a hope that given the rather restricted training and test material, there is still enough valuable information in the phone durations for them to be useful for testing the duration modeling techniques.

In addition to phonetic motivation, one point in studying the phone durations in the context of speech recognition is the fact that these durations are generally modeled very poorly due to use of HMMs. Their intrinsic state duration distribution is a geometric distribution, which is far from being suitable for its purpose. But being mathematically very attractive, it is the model most commonly used. It has also been noticed that the actual parameters for this poor duration model, that is, the transition probabilities of the HMMs, have very little effect to the recognition performance [4]. This is so, because the acoustics of the speech give by far the most

important clues for the recognition. But remembering the facts about phone durations in natural speech, it could be expected that better models would make a difference.

4.2 Duration distribution models

As have been mentioned several times, hidden Markov models have an intrinsic geometric distribution for its state durations. In phoneme based acoustic models each phone is modeled with its own HMM, usually consisting of three states. This already gives some freedom in modeling the phone durations. But as each HMM state has its own emission probability density and the paths over the HMM are always determined conditional to the acoustics, it is not easy to analyze the overall contribution of transition probabilities to the phone durations. And beyond, by the definition of Markov models, we are not able to model the correlations of durations between the states of HMM. Nevertheless, we can consider the joint duration distribution of the HMM states as a prior distribution for phone durations. If that prior is close to the real duration distribution, the realized durations should also be close to the objective.

The duration distribution of one HMM state was shown in Equation 2.17. The joint duration distribution of two HMM states, ignoring the correlation, can be derived via convolution, a discrete one in this case. The reasoning behind this is that we can permute over all the possible individual durations summing to certain joint duration. Thus, if the individual state duration distributions are $P_1(d)$ and $P_2(d)$, the joint duration distribution is

$$P(d) = \sum_{y=1}^{d-1} P_1(y) P_2(d-y). \tag{4.1}$$

We can now derive the distribution in case of HMMs, implying geometric duration distributions:

$$\begin{aligned}
P(d) &= \sum_{y=1}^{d-1} a_{11}^{y-1}(1-a_{11}) a_{22}^{d-y-1}(1-a_{22}) \\
&= \frac{(1-a_{11})(1-a_{22})}{a_{11}a_{22}} a_{22}^{d} \sum_{y=1}^{d-1} \left(\frac{a_{11}}{a_{22}}\right)^{y} \\
&= \frac{(1-a_{11})(1-a_{22})}{a_{11}a_{22}} a_{22}^{d} \left[\frac{\left(\frac{a_{11}}{a_{22}}\right)^{d} - 1}{\left(\frac{a_{11}}{a_{22}}\right) - 1} - 1 \right] \qquad , a_{11} \neq a_{22} \\
&= \frac{(1-a_{11})(1-a_{22})}{a_{11}a_{22}} \left(a_{11}^{d-1} - a_{22}^{d-1} \right)
\end{aligned} \tag{4.2}$$

For the second line, the terms independent of the sum variable are moved outside the sum and the coefficients are moved under the same exponent. For the third line, the sum of geometric series is applied, and the expression is then simplified for the last line. The result can be seen as a difference between two scaled geometric distributions.

The joint distribution has a close resemblance to the negative binomial, a discrete counterpart of the gamma distribution. This can be seen when considering the case where $a_{11} = a_{22} = a$:

$$P(d) = \frac{(1-a)^2}{a^2} a^d \sum_{y=1}^{d-1} 1^y = (d-1)(1-a)^2 a^{d-2}. \tag{4.3}$$

The result is now the point probability function of $Negbin(2, a)$.

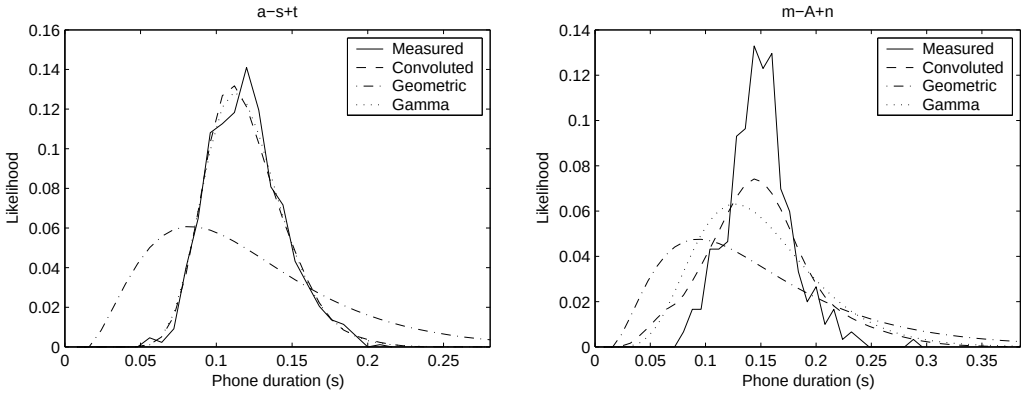
Another view for the joint duration distribution is seen when considering the means and variances of the distributions. From basic probability theory, the sum of independent random variables X and Y has properties:

$$E[X+Y] = E[X] + E[Y] \tag{4.4}$$
$$Var[X+Y] = Var[X] + Var[Y] \tag{4.5}$$

From the definition of HMMs, the durations of the states are indeed independent, and the joint duration can be seen as a sum of the individual durations.

The derived joint distributions above are for two-state HMMs. For the standard three-state case, the joint distribution can be computed by replacing the P_1 in Eq. 4.1 with the previous two state derivation. The results are similar, for the general case a combination of three geometric distributions, and for the reduced one parameter case the distribution is $Negbin(3, a)$. The general case has now three free parameters, but the question is, is the functional form of the joint distribution such that it can well reproduce the actual phone duration distributions? For this we need some measurements.

Figure 4.1 shows measurements of the durations of two context dependent phones (triphones) and their models. The notation of the figure titles means that Fig. 4.1a shows the durations of /s/ which has /a/ as the left context and /t/ as the right context. The capital letter in Fig. 4.1b represents a long Finnish phoneme. The measurements were made from the training data of the speech recognition tests reported in Chapter 5, using the segmentation generated by the training process. The about 12 hour material contained 879 samples of the first triphone (4.1a), and 301 samples of the second one (4.1b). The solid lines show the unsmoothed measured distributions of the durations of these triphones. The dashed curves are the convolutions of individual measurements of the three HMM state durations, therefore ignoring the correlations between the durations of the states. The curves being rather similar to



(a) (b)

Figure 4.1: Phone duration measurements and approximations for two context dependent phones. See the text for explanation.

the real measured phone durations show that ignoring the correlations between the durations of the HMM states do not mix up the analysis too much.

The dash-dotted curves are the convolutions of the three geometric distributions fitted to the individual durations of the HMM states. The results are significantly worse than the dashed curves indicating that three-state HMMs with intrinsic geometric duration distributions are unable to model the phone durations correctly. If, however, the geometric durations are replaced with some more flexible distributions, better results emerge. In the figures there are also dotted curves, which were computed otherwise the same way as the dash-dotted ones, but using gamma distributions instead of the geometric distributions (more correctly, the used distribution should be the negative binomial as we actually deal with discrete durations, but gamma distribution is used for computational convenience). The fit is now much better, especially in Fig. 4.1a, where it almost perfectly fits with the convolution of the measured state durations. The Fig. 4.1b shows, however, that the superiority is not always that substantial.

The parameters of the duration distributions were optimized to fit the individual state durations, not the phone durations (which could be either measured or convoluted). One can therefore argue that the exponential distributions could perform better if fitted in some other way. But it is not at all clear how the optimization should be done to achieve this. Furthermore, Equations 4.4 and 4.5 suggest that the overall distribution will always have strong restrictions, as the mean and variance of the geometric distribution are closely coupled due to single parameter. Optimizing the HMM parameters as a whole also blurs the correlations and assumptions contained by the model, especially when taking the conditional acoustic probabilities into account. Using gamma distributions seems to avoid the problem, as it produces good fits even

when optimized simply state by state.

Some attempts for altering the parameter optimization schemes have still been developed. For example, [14] describes a procedure for constraining the individual state duration variances to produce desirable phone duration variances. However, it was made using Gaussian duration distributions, and the constraint was mainly used to decrease the HMM duration variance from what the measured durations indicated, as this was empirically found to improve the recognition results. The reduction of variance is in fact in close resemblance to the scaling of the different likelihood values described in Chapter 3, as both of these methods result in more "spiky" likelihood values.

Estimating gamma distribution parameters

The previous considerations suggest that the gamma distribution is well suitable for modeling the state durations of HMMs. But what we have not yet discussed is how to estimate the parameters for this or any other choice of distribution, to fit them to the duration distributions of each HMM state. The usual way of estimating distribution parameters is to maximize the likelihood of the training data, therefore the name Maximum-Likelihood (ML) estimate. It is somewhat more justified method than for example the method of moments, which fits the mean and variance of the distribution and data to be the same. For geometric distribution, the method of moments, which in that case reduces to fitting the mean of the distribution to that of the data, equals to the ML estimate, but this is not the case with the gamma distribution.

The ML estimate for the gamma distribution can be derived as follows [1]. The gamma distribution has two parameters, and its probability density function is

$$f(x;a,b) = \frac{x^{a-1}\exp(-\frac{x}{b})}{b^a\Gamma(a)}. \tag{4.6}$$

The likelihood function to be maximized is

$$L(a,b) = \prod_{i=1}^{N} f(x_i;a,b) = \frac{1}{(b^a\Gamma(a))^N}\prod_{i=1}^{N} x_i^{a-1}\exp(-\frac{x_i}{b}) = \frac{\exp(-\frac{1}{b}\sum_{i=1}^{N} x_i)}{(b^a\Gamma(a))^N}\prod_{i=1}^{N} x_i^{a-1}, \tag{4.7}$$

where N is the number of data points and x_i is the data set (durations). The maximization is simplified by taking a logarithm of the likelihood function. This leads to

$$\log L(a,b) = -Na\log b - N\log\Gamma(a) - \frac{1}{b}\sum_{i=1}^{N} x_i + (a-1)\sum_{i=1}^{N}\log x_i. \tag{4.8}$$

[1] Based on the derivation found from:
http://www-mtl.mit.edu/CIDM/memos/94-13/subsection3.4.1.html

If this expression is now derivated and the derivatives are set to zero, we end up in an equation which we can not solve analytically. Thus we can as well maximize the log likelihood directly using some numerical method. However, by derivating the above expression relative to b, we get additional constraint to the maximization and can transform the problem to a maximization over single variable. The derivative gives

$$\frac{\partial \log L}{\partial b} = \frac{-Na}{b} + \frac{1}{b^2}\sum_{i=1}^{N} x_i = 0 \quad \Longleftrightarrow \quad b = \frac{1}{Na}\sum_{i=1}^{N} x_i. \tag{4.9}$$

Substituting this into Equation 4.8 finally gives the function to be maximized numerically:

$$\log L(a) = -N\left(a \log\left(\frac{1}{Na}\sum_{i=1}^{N} x_i\right) - \log\Gamma(a) - a + \frac{a-1}{N}\sum_{i=1}^{N} \log x_i\right). \tag{4.10}$$

The resulting function is rather easy to be maximized iteratively, for example, with a simple golden ratio method [33]. Good initial values can be found with the method of moments. With the above parameterization, the mean of the gamma distribution is ab and the variance is ab^2. Once the correct a parameter is found using the iterative maximization, the b parameter can be computed from Equation 4.9. This also implies that the gamma distribution estimated in a maximum likelihood manner has the same mean as the data (due to the way the b parameter is computed) but the variance may differ.

As we now have a motivation for modeling the phone durations well, and a suggestion for a duration distribution managing to do this, we are ready to begin to investigate how this modeling can be done in the framework of HMM algorithms used in speech recognition.

4.3 Hidden semi-Markov models

Perhaps the most straightforward solution for altering the state duration distributions with HMMs is to explicitly define the duration distributions to the HMM formalism. This extends the models to be hidden semi-Markov models [34].

Referring to the formal definition of HMMs in Section 2.2, a hidden semi-Markov model (HSMM) can be described as a tuple $\{S, A, B, \pi, D\}$, where $S = \{s_j\}$ is the set of states in the HMM, $A = \{a_{ij}\}$ is the probabilistic transition matrix, $B = \{b_j(\boldsymbol{o})\}$ is the set of emission PDFs, $\pi = \{\pi_j\}$ is the initial state distribution and $D = \{d_j\}$ is the set of duration PDFs. To simplify the analysis, we restrict the transition matrix to have no self transitions, that is, $a_{ii} = 0$ for all i. The functional difference to the normal HMM is that the occupancy of a state is not defined by the transition matrix

(that is, the self transitions a_{ii}) but rather an explicit state dependent duration PDF. When in a HSMM a transition to the state s_j is made, it is occupied for a time according to the state's duration distribution d_j, after which a new transition is made, according to the transition matrix A.

The definition of the HSMM violates the Markov assumptions introduced in section 2.2. This is because at any time, the transition probability (the self transitions with normal HMMs correspond to the lack of transitions with HSMMs) depends on how long the process has remained in the current state. This implies that the same effect as with HSMMs can be achieved by defining the transition probabilities of the underlying Markov chain to be dependent of the time occupied by the state. This kind of approach is sometimes called inhomogeneous hidden Markov model [32].

The computational advantage of the Markov assumption becomes very clear when formulating the Viterbi algorithm for both normal HMM and HSMM. The Viterbi algorithm is at the heart of a speech recognizer; it is used to find the best state sequence over the given HMM. Without explicit duration probabilities, this can be defined as a fairly simple recursive procedure [31]. Let us denote with $\delta_t(i)$ the probability of the best state sequence ending in state s_i at time t. If we initialize this quantity as

$$\delta_1(i) = \pi_i b_i(\boldsymbol{o}_1), \quad 1 \leq i \leq N, \tag{4.11}$$

we can define the recursive procedure as

$$\delta_t(i) = \max_{1 \leq j \leq N} \{\delta_{t-1}(j) a_{ji}\} b_i(\boldsymbol{o}_t), \quad 2 \leq t \leq T,\ 1 \leq i \leq N, \tag{4.12}$$

where N is the number of HMM states, $\boldsymbol{o}_t$ is the acoustic feature vector at time t and T is the last time index of the sequence. We can then find the best state sequence by finding the best probability of state sequences ending at time T, i.e.

$$P^* = \max_{1 \leq i \leq N} \{\delta_T(i)\}, \tag{4.13}$$

and backtracking the states through which the recursion passed for that best state sequence.

It should be noted that the recursion in Equation 4.12 can be computed very efficiently. At each time instant the algorithm only needs to know the probabilities of the best state sequences ending at each HMM state at the previous time instant to update these probabilities. Furthermore, the topological restrictions of the HMMs can be utilized easily to speed up the computation. For example, for the simple left-right model which was shown in Figure 2.4 the maximization in Eq. 4.12 only has to consider indices $j = i - 1$ and $j = i$, that is, the transition from the preceding state and the self-transition. With this restriction, the overall time requirement of the algorithm is only $O(NT)$.

To extend the Viterbi algorithm for HSMMs, it has to be modified slightly, but unfortunately at the expense of efficiency. Following the notations in [3] (with some

small modifications), it is now better to define $\delta_t(i)$ to be the probability of the best state sequence ending in state s_i at time t, but which will be in another state at time $t+1$. This way, the recursive update equation becomes

$$\delta_t(i) = \max_{t-D\leq\tau<t} \left\{ \max_{1\leq j\leq N} \{\delta_\tau(j)\, a_{ji}\}\; d_i(t-\tau) \prod_{k=\tau}^{t} b_i(\boldsymbol{o}_k) \right\}, \tag{4.14}$$

where D is the maximum duration of a single state. It is easy to see that the evaluation of this recursive declaration is much more demanding than the one obtained for normal HMMs. As the state of the model no longer depends only on the state at the previous time instant, the algorithm has to consider D different durations for each HSMM state, resulting in a slow down by factor D. A reasonable value for D is of order 25 [31], so the efficiency of the algorithm is seriously compromised. Additionally, more storage room is also required, as each state of the model has to be accompanied by the duration how long the model has remained in that state.

Initially, non-parametric duration distributions were used with HSMMs. That is, each $d_i(\omega)$ were estimated for all $1 \leq \omega \leq D$. But this way, vast amounts of training data was needed to estimate all the parameters well. Therefore several parametric duration distribution functions have been incorporated with the HSMM and explicit re-estimation formulas have been derived. The distributions used with HSMMs include, for example, Poisson PDF [34], gamma PDF [22], and Gaussian PDF [31].

Speeding up the HSMMs

The slow down by factor D is not entirely intolerable, but more efficient solutions, even with some simplifications, would be desirable. Several suboptimal algorithms incorporating duration penalties inside the traditional Viterbi algorithm with update rule similar to (4.12) has been presented, for example in [32, 43, 6]. These have been reported to improve the recognition when compared to the standard HMMs, and the algorithms suffer virtually nothing in efficiency. But being suboptimal, they do not account for real HSMMs.

Bonafonte *et al.* [3] presented a pruning theorem with which the search space for Equation 4.14 can be significantly reduced without compromising the optimality of the algorithm. They reported an increase of computational effort of about 3.2 times with respect to conventional HMM, the increase being almost independent of the actual value of D. This kind of decrease in performance can be tolerated easily, and that method is the one adopted to be used in this thesis.

The derivation of this pruning theorem (adapted from [3]) is as follows. Suppose the best path leaving state s_i at time t had left previous state s_j at time τ_1. We

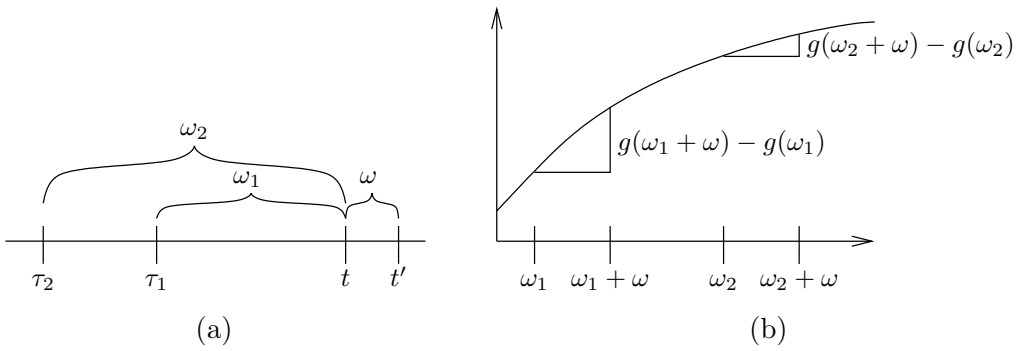


Figure 4.2: Illustration of the time and duration notations (a) and an example of a logarithm of a proper duration probability function with monotonous behavior (b), relevant to the pruning theorem of HSMMs.

then examine another path which moved from state s_k to state s_i at time τ_2 and ask ourselves under which condition the first path remains to be the best when considering $\delta_{t'}(i)$, where $t' > t$. From the knowledge that the first path was the best at time t we know that

$$\frac{\delta_{\tau_1}(j)a_{ji}\prod_{l=\tau_1+1}^{t} b_i(\boldsymbol{o}_l)\ d_i(t-\tau_1)}{\delta_{\tau_2}(k)a_{ki}\prod_{l=\tau_2+1}^{t} b_i(\boldsymbol{o}_l)\ d_i(t-\tau_2)} = K > 1. \tag{4.15}$$

These sub-paths can then be compared when considering $\delta_{t'}(i)$:

$$\frac{\delta_{\tau_1}(j)a_{ji}\prod_{l=\tau_1+1}^{t'} b_i(O_l)\ d_i(t'-\tau_1)}{\delta_{\tau_2}(k)a_{ki}\prod_{l=\tau_2+1}^{t'} b_i(O_l)\ d_i(t'-\tau_2)} = K \cdot \frac{d_i(t'-\tau_1)/d_i(t-\tau_1)}{d_i(t'-\tau_2)/d_i(t-\tau_2)} \overset{?}{>} 1. \tag{4.16}$$

The most critical case is when $K \approx 1$. In this case, let us compare the sub-paths by taking logarithms of the previous inequality. To simplify the notation, define $g_i(x) = \log(d_i(x))$. Denoting $\omega_1 = t - \tau_1$, $\omega_2 = t - \tau_2$ and $\omega = t' - t$, we get:

$$g_i(\omega_1+\omega) - g_i(\omega_1) \quad \overset{?}{\geq} \quad g_i(\omega_2+\omega) - g_i(\omega_2). \tag{4.17}$$

If $g_i'(\omega_1) > g_i'(\omega_2)$ and $g_i'(x)$ is monotonous in an interval containing ω_1,$\omega_1+\omega$, ω_2 and $\omega_2+\omega$, then the inequality 4.17 is true. Figure 4.2 illustrates these concepts.

We can now deduce that the inequality holds in the following cases:

- $\omega_1 > \omega_2$ and $g_i'(x)$ is monotonically increasing, i.e. $g_i''(x) \geq 0 \ \ \forall x$
- $\omega_1 < \omega_2$ and $g_i'(x)$ is monotonically decreasing, i.e. $g_i''(x) \leq 0 \ \ \forall x$

From now on, we restrict to the latter of these cases. A function is said to be log-convex if it obeys $[\log(p(x))]'' \leq 0 \ \ \forall x$. With this and previous definitions, we can state the pruning theorem as follows:

> If the state duration PDF of the state i is log-convex, then if the best path leaving state i at time t has arrived at state i at time τ from state j then the best path leaving state i at time $t+1$ has arrived at state i at
>
> - time τ from state j
> - time τ' from any state, with $\tau' > \tau$

This theorem allows us to limit the search over the best τ in Equation 4.14 to be at most the duration of the best path at the previous time instant. This is usually much smaller than D, thus speeding up the algorithm considerably. Moreover, most parametric functions used by HSMMs are log-convex [3], enabling us to utilize this pruning theorem.

HSMMs and gamma distributed state durations

In section 4.2 gamma distribution was noted to be able to model the state durations well. Let us now consider under which constraints it is log-convex, and therefore applicable to be used with HSMMs and the pruning theorem presented above.

The probability density function of gamma distribution was presented in Equation 4.6. Computing a logarithm gives

$$\log f(x;\ a, b) = (a-1)\log x - a\log b + \log\Gamma(a) - \frac{x}{b}. \tag{4.18}$$

Taking the derivatives with respect to x and requiring the second derivative to be negative results:

$$[\log f(x;\ a, b)]' = \frac{a-1}{x} - \frac{1}{b} \tag{4.19}$$

$$[\log f(x; a, b)]'' = \frac{1-a}{x^2} \leq 0 \quad \Longleftrightarrow \quad a \geq 1 \tag{4.20}$$

This constraint should be taken into account when solving the Equation 4.10. However, the constraint does not restrict the gamma distribution in an important way when considering the modeling of state durations.

As was mentioned before, distribution parameter re-estimation schemes have been presented for various distributions useful in modeling the state durations of HSMMs. These incorporate the optimization of the parameters inside the EM-algorithm used to train the acoustic models. In this work, a slightly different and more general approach was taken. Instead of writing explicit re-estimation formulas inside the training algorithms, durations statistics for the HMM/HSMM states are collected during the training and the duration parameters are trained afterwards, using ML estimation. As the training of the acoustic models is an iterative process and the duration distributions are estimated after each iteration, the distribution parameters should converge correctly, as was empirically noticed.

Implementational issues with HSMMs

Implementing HSMMs in a speech recognition system requires modifying the existing Viterbi search to incorporate the duration probabilities and to take into account the complicated dependencies. For a HMM state at a given time instance, it is no longer enough to store only which transition just took place, but a history of entrance times and corresponding likelihood values from earlier time instances (up to D) is needed. The memory requirements of the algorithm increase, but this should be insignificant compared to the memory requirements of the acoustic and language models. Besides, if the pruning theorem presented by Bonafonte *et al.* is implemented, it is rarely necessary to store all the time instances up to D, and therefore it is possible to benefit from dynamic memory allocation.

The running time of the Viterbi search and therefore the overall decoding efficiency is highly dependent on the pruning of the improbable paths. When using normal HMMs, it is effective to leave those HMM states outside of path consideration for which the probability of occupying that state at some time instance deviates too much from the probability of being in the best state at that same time instance. This is the beam parameter mentioned in Section 3.2. More precisely, at time t state s_i is not considered to possibly belong to the overall optimal path if

$$\delta_t(i) < \delta_t(k) - beam, \tag{4.21}$$

where $\delta_t(k) > \delta_t(j)$ for $\forall j$ and $beam$ is the beam parameter scaled according to the state probabilities, which are actually represented as likelihoods.

However, with HSMMs this kind of pruning is no longer intuitively that appealing. Unlike with normal HMMs, there are no self transition probabilities to penalize staying in the same state. This penalty is done in the form of duration probabilities, but they are applied only when changing the state. Thus this kind of pruning is likely to overestimate the probability of a state which has been occupied for a while. It may therefore be necessary to run the Viterbi search with more relaxed beam values, which reduces the decoder efficiency even further. It could be of course possible to

devise more complicated pruning schemes which would compensate for the missing duration penalty while staying in the same state. In preliminary tests a couple of these were implemented, but none of those seemed to improve the recognition performance.

HSMMs with the pruning theorem described above are in a way optimal models for incorporating duration distributions into the HMM framework. That is, no additional assumptions or approximations are made in the computations with the models. This makes it appealing to use it also in the training part of speech recognition. The speech recognition system utilized for this study uses Viterbi algorithm also in training, although with different implementation than in the decoder. It was therefore rather straightforward to implement HSMMs also to the training phase to see its effect. A comparison of the recognition results using the models obtained with and without training are shown in Chapter 5.

4.4 Expanded state HMM

It is well known that Markov chains can model general probability distributions [8, 9]. As the acoustic models already rely on Markov chains, why would not use this ability of modeling probability distributions to better model the durations? The idea is to expand each HMM state to a sub-HMM, which shares the same emission probability density and realizes the correct state duration distribution with its topology and transition probabilities. This kind of model is called the expanded state HMM (ESHMM) [35].

When implementing such a model, it is very important to note under which conditions the model is used. The low level speech recognition procedure finds the single best path over the HMM using the Viterbi algorithm. If we then rely on Equation 4.1 to produce the duration distribution over the HMM, we will fail due to fact that we are not summing over all the possible paths [35]. The analysis carried out in section 4.2 worked because there we considered paths over all the occurrences of phones in the speech material, each conditioned with acoustic information and emission probabilities of HMM states. But if we are to realize the plain duration distribution during the Viterbi algorithm over sub-HMMs, we need to consider the probabilities formed by the best paths for each duration, without being conditional to the acoustic information. This is so because the states of one sub-HMM share the same emission probability density. Fortunately, it is still possible to form HMM topologies which allow free modeling of such durations.

For the Markov chain to represent a duration distribution with Viterbi algorithm, the best paths (determined by the transition probabilities) realizing each duration must be clearly defined. The easiest way is to define explicit transitions which are

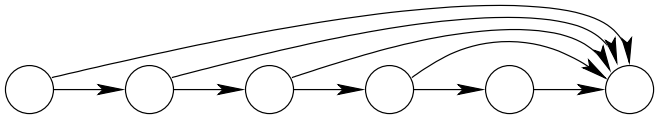


Figure 4.3: Fergusson topology sub-HMM.

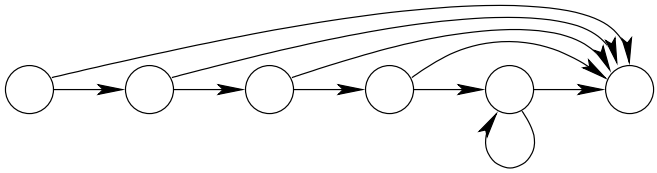


Figure 4.4: More general topology for sub-HMM.

used only for certain durations. This kind of topology, as shown in Figure 4.3, was presented in [35], and it is there referred as Fergusson topology. The name of the model originate from the fact that this kind of topology can model arbitrary distributions up to its length, corresponding to the first suggested HSMMs with non-parametric distributions [22], so called Fergusson model.

The problem with the Fergusson topology is that the longest duration it can model is determined by the number of states in the sub-HMM. For the maximum duration of 25 time steps per HMM state discussed with HSMMs, this would require vast amounts of sub-HMM states. But if a self transition is introduced in the second last state of the sub-HMM, the tail of the duration distribution can be modeled as a geometric distribution, like the state durations in the standard HMMs. This was suggested in [27]. However, in that paper, the transition probabilities of the sub-HMM were fixed to be the same, therefore limiting the forms of distributions this model could represent. In this work, the same topology as in [27] is used for ESHMMs, but the parameters are not bound explicitly. The sub-HMM topology of the adopted model is shown in Figure 4.4. As the topology of sub-HMMs is very restricted and the states of the sub-HMM corresponding to one original HMM state share the same emission probability density, the increase of computational effort is moderate and less than that with HSMMs.

Duration distribution estimation for ESHMM

If the state durations are modeled using ESHMMs, there are many decisions to be made to fix the parameters of the models. At first, the correct number of states for each sub-HMMs has to be determined. Then, one could in principle train the ESHMMs using Viterbi training to obtain ML estimates for the transitions. But

now the same problem arises as with the non-parametric Fergusson model HSMM: The number of free parameters requires huge amounts of training data. To alleviate this, the transition probabilities of sub-HMMs should be constrained in some way to produce smooth duration distributions even with smaller number of training samples.

Before the transition parameters can be estimated, the number of states in each sub-HMM has to be decided. This number can be some fixed value common for all sub-HMMs as in [35], but in this work a data driven method was used to select the number of states for each sub-HMM. The procedure was coupled with the estimation of the transition parameters. For each HMM state duration statistics were gathered in the training phase, as was done with the training of HSMMs. Then for each HMM state different lengths of sub-HMMs were tested in order to find the optimal fit to the measured duration distribution. The goodness of fit was determined with Kullback-Leibler distance [2] between the measured and modeled distributions. The distance measure was penalized with empirically scaled values of probability mass accurately modeled by the sub-HMM (that is, the probabilities of durations less than those involving transition to the second last sub-HMM state with the self-transition) and the number of states in the sub-HMM, so that a threshold could be defined for the optimal fit. When the number of states was selected for which this penalized distance was just below zero, good looking fits emerged. The empirical values for the penalty coefficients when fitting to the state durations of triphones were 0.07 for the probability mass and 0.004 for the number of states.

To alleviate the problem of being short of the training data, the measured state duration statistics were smoothed before the sub-HMM parameters were fitted. Two smoothing schemes were experimented. In the first method, the duration distribution was convoluted with a Gaussian window of length 5. This is so called kernel method [41], which can be used to make a spiky density estimate more robust. In the other method, a gamma distribution was fitted to the measured state durations and sub-HSMMs were fitted against it. This way tests comparable to the other duration modeling techniques could be made. Figure 4.5 shows example fits with these two methods to the measured HMM state duration. In this example both methods found their optimum fit by using three explicit duration states and a state with self transition.

Let us finally consider the actual estimation of the transition parameters. Once the topology and the number of states has been decided, one could, of course, use the traditional Viterbi training to achieve the ML estimate for the parameters. But especially for the optimization procedure described above, it is advantageous to be able to set the parameters explicitly. This is in fact rather easy with ESHMM shown in Figure 4.4. The probability of unit duration is determined by the transition probability from the leftmost state to the last state. Let us denote this with k_1. The probability of duration $d(2)$ is now determined by a transition to the second state multiplied by the transition to the last state. If the latter is denoted as k_2,

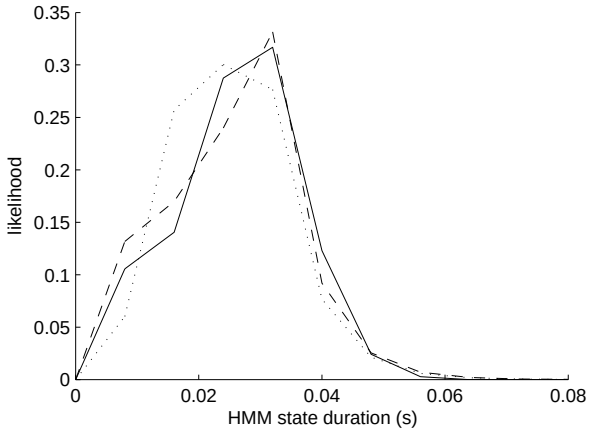


Figure 4.5: Two fits to the HMM state duration using ESHMMs. Solid line shows the measured duration, dashed line is the Gaussian smoothed fit and dotted line is the gamma fit.

the probability can be described as $d(2) = (1 - k_1)k_2$. Proceeding this way, we have $d(3) = (1 - k_1)(1 - k_2)k_3$ and so on. When the second last state with the self-transition is reached, we only have to estimate that last parameter of geometric distribution, for example with an ML estimate of fitting its mean to the mean of the remaining probability distribution.

Using ESHMMs in a speech recognizer

As ESHMMs are normal HMMs merely constructed in a specific way, their use in a speech recognition system is very easy, as long as the system supports HMMs of different lengths and tied emission densities. With the Viterbi algorithm, the proper scaling of transition probabilities is important, as they constitute the actual duration information. Otherwise no special care has to be taken, as long as it is realized that the number of HMM states increases with the ESHMMs. This can be an issue in some pruning strategies.

In principle ESHMMs can be used in the training part of the speech recognizer as well as in decoding. However, when constructing the ESHMMs from the training data, the number of states in sub-HMMs is fixed. Therefore the training can only adapt the transition probabilities of these sub-HMMs. If some kind of smoothing is applied for the duration distributions, which is highly recommended, the same smoothing has to be applied after each training iteration, as EM-algorithm fits the sub-HMM transitions to model exactly the measured state durations up to sub-HMM length.

4.5 Post-processor duration model

In addition to the two duration modeling techniques described in the previous sections, an even simpler approach exists. It is not mathematically as well justified, but the advantage is that it has practically no impact to the efficiency of the recognition algorithm. This method is called the post-processor duration model [17]. It uses exactly the same Viterbi algorithm as is used with normal HMMs, presented in the Equation 4.12, but it ranks the resulting paths according to how well the paths fit in to the duration distributions measured at the training time.

The post-processor method can be derived from the following probabilistic consideration. The decoder of a speech recognizer is given models for acoustic and duration models, denoted by λ_a and λ_d, respectively. If simplified, the decoder can be seen to pick the phoneme sequence W for which the likelihood $p(\boldsymbol{O} \,|\, W, \lambda_a, \lambda_d)$, where $\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_m$ is the sequence of acoustic feature vectors, is the highest. This can be written as a sum over the paths of the HMM forming the phoneme sequence W:

$$p(\boldsymbol{O} \,|\, W, \lambda_a, \lambda_d) = \sum_{Q_i \in Q} p(\boldsymbol{O} \,|\, Q_i, \lambda_a) P(Q_i \,|\, W, \lambda_d), \tag{4.22}$$

where Q is the set of valid HMM state sequences. In the Viterbi algorithm, the likelihood is approximated with a single best path:

$$p(\boldsymbol{O} \,|\, W, \lambda_a, \lambda_d) \approx p(\boldsymbol{O} \,|\, Q_{best}, \lambda_a) P(Q_{best} \,|\, W, \lambda_d). \tag{4.23}$$

The first term of the right hand side is simply the acoustic probability given the state sequence Q_{best}. In the context of HMMs, the second term is the product of transition probabilities for the state sequence Q_{best}, and this is just how the Viterbi algorithm evaluates it. However, this term could be computed more correctly if we forget the HMM context and adopt more accurate state duration models.

The idea behind the post-processor duration model is thus as follows: With the Viterbi algorithm we are able to find good paths through the HMMs forming different phoneme sequences. After obtaining the paths, it is easy to recalculate the probabilities of the paths using the Equation 4.23 with better duration models than the standard HMM offers. The assumptions are that we get the correct paths from the Viterbi search and ranking them using better duration models really gives us more information to aid choosing the best hypothesis. The evident problem results from the former assumption, as the best path relative to the better duration models need not be the same as the one Viterbi algorithm finds using the simple geometric duration model. However, practice has shown that the post-processor method does perform very well despite this troublesome inconsistency.

The original version of the post-processor model, presented in [17] and also in [31], has a slightly more heuristic interpretation. Instead of evaluating the likelihoods

with better duration models, an additional penalty is suggested to be added to the log-likelihood resulting from the Viterbi search. This results in an equation

$$\log \hat{P}(\boldsymbol{O} \,|\, W, \lambda_a, \lambda_d) = \log \left(p(\boldsymbol{O} \,|\, Q_{best}, \lambda_a) P(Q_{best} \,|\, W, \lambda_d) \right) + \alpha \sum_{i=1}^{N} \log d_i(\tau_i), \quad (4.24)$$

where α is an empirical scaling factor, the sum is taken over the states of the HMM and τ_i is the time path Q_{best} spends in the HMM state s_i. The difference is thus that the original (log-)likelihood computed using geometric duration model is also taken into the final likelihood. In practice this has only minor implications.

The performance of the post-processor duration model depends on the actual implementation of the decoder in the speech recognizer. If the whole recognition is done with one hypothesis iteration as described above, the improvement achieved by using the post-processor model is probably small. But this kind of implementation is used only in the most simple connected-word speech recognizers. In the modern continuous speech recognizers, the decoder continuously generates and evaluates small hypotheses to proceed with the recognition. Usually several competing paths or hypotheses are kept as a starting point for new hypotheses, thus more path alternatives are presented to the post-processor model. This makes it more probable that the correct paths are evaluated with Equation 4.23 using the better duration models and so ranking the new hypotheses more correctly and guiding the recognition to more accurate results. To make the post-processor method even more justified, one could use the so called N-best paradigm [37] to get several path alternatives to be evaluated instead of the single best obtained from the traditional Viterbi search.

It should be emphasized that with both post-processor implementations, the local Viterbi search which the decoder performs to expand the existing hypotheses remains untouched, the duration penalty is only added to the likelihoods of the expanded hypotheses. Thus the decoder's beam parameter has no direct control over the post-processor model, other than with larger beam values the Viterbi search ends up finding more morph alternatives which are then ranked with the explicit duration model.

Apart from the empirical scaling factor α shown in Equation 4.24, it may be again necessary to scale the transition probabilities too, to get the best paths from the Viterbi search. However, when implemented in the form of Equation 4.24, the transition probability scaling degraded the decoder performance slightly. An unfortunate drawback of the post-processor model is that it is not possible to use it in the training phase to improve the acoustic models.

4.6 Speaking rate adaptation

The speaking rate is probably the most easily observable source of phone duration variation. In addition to the effect on durations, speaking rate may also affect the acoustic properties and pronunciations of the speech [36]. It is therefore no wonder that a mismatch between the speaking rate of the training material and the recognized speech has been reported to degrade the accuracy of the recognition [38, 26]. Especially the recognition of fast speech seems to be problematic [36].

As this thesis concerns mainly the durations of the phones, only that part of the speaking rate is studied here. However, no implementations of algorithms measuring or utilizing speaking rate were made within this thesis. The consideration is here only to enlighten the effect and use of speaking rate with the duration models presented in this chapter.

The actual definition of the speaking rate is not so straightforward. The word rate and the phone rate have been suggested for the use, and of these the latter has been shown to be better for speech recognition applications [38]. But what still remains to be defined is the interval over which the rate is computed. Instantaneous phone rate has been defined to be the inverse of the phone duration, and mean phone rate to be the arithmetic average of that over some interval, for example, the entire utterance [38]. However, the speaking rate may vary even inside the utterance [44], so also shorter intervals have been used [26].

To compute the phone rate exactly, a transcription is required. If the speaking rate measure is needed on the recognition time, the decoder needs to run multiple passes to first obtain a preliminary transcription from which the rate is estimated, and then perform the actual recognition [36]. Because this kind of approach is computationally very intensive, several methods for estimating the speaking rate prior to having the transcription available have been developed. A rather straightforward method is to estimate the phone rate from the hypothesized transcription obtained so far [38]. Other methods include estimating phone boundaries with multi-layer perceptrons (MLP) [44], estimating the speaking rate from the spectral changes of speech using Gaussian mixture models [10], analyzing the energy envelope fluctuations of the speech signal [26] and several others [36]. Without getting too deeply in to the topic, these on-line measures have been proved to be useful in estimating the speaking rate for various applications.

After an estimate have been obtained for the speaking rate, one would wish to use it to improve the recognition accuracy. For duration models, this involves either selecting appropriate duration models trained for particular speaking rate as in [44], or adapting global duration models for the particular speaking rate [36]. The duration model available for adaptation clearly affects the effectivity. In [26], the geometric distribution implicitly defined by HMM transition probabilities was adapted for

different speaking rates. It was noted there that explicit duration models would benefit the adaptation more, because the duration distributions of rapid speech were much more skewed than those of slow speech. Unfortunately it is not easy to derive how the duration distributions should be adapted with respect to the speaking rate. The effect of the speaking rate to the durations of phones depends on the stress. Furthermore, the durations are affected in a non-linear way [16]. However, a linear compensation has been reported to reduce the variance of the expected vowel durations [36].

In the view of the duration modeling techniques presented in this chapter, we should also consider how speaking rate adaptation could be implemented inside these methods, if reasonable estimates and duration statistics for achieving this would be available. For HSMM and post-processor duration model this adaptation would be rather easy, as they only had to update parameters of their statistical duration models. A minor complication is where the duration distributions are actually switched from one to another, but this should not have much of an importance if the switching is done consistently. On the other hand, modeling durations using ESHMM would pose more difficulties as the actual HMM models had to be changed. Depending on the implementation, this might only require computationally easy recomputation of the transition probabilities, but with another kind of architecture, implementing the adaptation might turn out to be very difficult.

Chapter 5

Experimental Evaluation

5.1 Test setup

To evaluate the described duration modeling techniques, speech recognition tests were performed. All the techniques were evaluated with the same speech material, which was a Finnish book spoken by one female reader. An extract of 12 hours was used to train the acoustic models, and independent parts of 9 and 30 minutes were used as development and evaluation sets, respectively. Speaker dependent models are justified in order to reduce the unwanted variations in phone durations which are not modeled in the current system. A continuous speech recognition task requires a language model, so a trigram language model was used, trained with a separate text corpus consisting of about 30 million words. The language model was based on morphs, as explained in Chapter 3.

The tests were evaluated using triphone models, which are the usual acoustic models with modern speech recognizers. Triphones should in some extent compensate for the effect of phoneme context to the phone durations. However, the training material restricted the selection of triphones so that a large number of somewhat more rare contexts were left out, and these were modeled using diphone or monophone models, depending on the number of occurrences in the training material. The models had 316 triphones, 296 diphones, 49 monophones and two models for silence (a short and a long one), constituting 663 separate HMM models in total.

In the acoustic models, single and double phonemes corresponding to short and long phones had separate HMM models. The baseline system, however, did not distinguish between these two variants acoustically, but left the decision of the correct variant of the phoneme entirely to the language model. Due to this difference, a separate set of HMM models was needed for the baseline system, which ended up having 629

phone models in total.

The speech recognition experiments performed involved recognizing the test material as if it was unknown speech, and then comparing the recognition result to the known transcription. For the actual evaluation, letter error rate (LER) was selected as the recognition accuracy criterion. This is in contrast to the usual choice of word error rate (WER) to the task. However, word error rate is not that well applicable for Finnish where long words consisting of many morphemes are common. A minor error in one letter of one morph would classify the word as misrecognized, although human reader of the recognized text could easily recover that kind of error. Word error rate also penalizes too much for misrecognized word breaks, which is an issue due to use of morphs as the language model units. Letter error rate, on the other hand, represents a smooth and meaningful criterion for recognition accuracy. It measures the relative amount of modifications (insertions, deletions and substitutions of letters) required for the recognized text if it is to be manually corrected. The actual task for which the speech recognizer is designed to naturally determines the proper error measure, and the arguments promoting the use of LER have been seen as the most relevant in the current system. Nevertheless, also WER measures are presented for reference.

Different modifications to the speech recognition system and especially to the decoder can alter the performance of the system in a number of ways. The two parameters measured in the experiments were the recognition accuracy and efficiency. The former is expressed by the LER, as noted in the previous paragraph, and the latter is measured by a real-time factor. A real-time factor is the time spent on the decoding (recognition) divided by the duration of the speech material, and it describes the effort the decoder does when recognizing the speech. This kind of measure is needed for fair comparison of different techniques affecting the decoder performance in various ways. The number of model parameters, the memory consumption, the computational complexity and pruning strategies all influence the running time of the decoder, and are therefore measurable via the real-time factor.

To be able to compare the different modeling techniques more precisely, the tests were run at different running times, controlled with the decoder's beam parameter. The results are given by reporting the LER as a function of the real-time factor. From this the behavior of the different modeling techniques becomes more clear. The actual real-time factors should, however, be interpreted only as relative values, as the tests were run with a rather old computer (550 MHz Pentium III) and the decoder system has not been optimized for speed.

Test for statistical significance

Whatever the recognition accuracy measure is used, the measurements itself are inherently noisy. Different test materials may be of varying levels of difficulty for

reasons like the quality of speech or the demandingness of the transcription. Comparing different models in speech recognition becomes difficult because one model may fit better to some kind of test material while another model may outperform others with some other material. For example, if one acoustic model is designed to be robust against noisy speech, it could obtain the best recognition accuracy among different models if tested with material where such robustness is needed, but with clear speech the other models may get the best recognition results.

To minimize the variation between the recognition accuracy measurements, all the duration modeling techniques were evaluated with the same test material, which has a good speech quality. Still it is possible that some methods get more positive results just because they fit better to the particular test material. To completely rule out this possibility, a huge amount of different kinds of test material would be required. This was not an option due to time and resource limitations. Fortunately it is also possible to analyze statistically how the different methods compare to each other. This way we get the measurement results with our restricted test material and also a significance measure about how confident we are that the differences between the measurements are indeed due to the better recognition techniques and not because of random measurement noise.

The statistical test used for evaluations is called a matched-pairs test [12]. It suits for test material which can be divided to independent segments. No assumptions are made on the independence of the recognition algorithms or on the distribution of the accuracy measurements. The idea of the test is to compute differences between the accuracy measurements of the two recognition techniques over each independent segment. Because these differences should have emerged from the same probability distribution, averaging them results in an estimate whose distribution approaches the normal distribution. We can normalize this estimate using an estimate for the variance of the mean. If $\hat{\mu}_Z$ denotes the estimate for the mean of the accuracy differences, $\hat{\sigma}_Z^2$ denotes the estimate for the variance of those same differences and n is the number of segments over which these estimates are evaluated, we can define a measure useful for analyzing the statistical significance:

$$W = \frac{\hat{\mu}_Z}{(\hat{\sigma}_Z/\sqrt{n})}. \tag{5.1}$$

Now as the distribution of W is approximately the standardized normal distribution, we can use its values to test the significance of the difference in the accuracy measurements using standard significance testing, which gives us the so called P value [25]. If we expect our modification to the recognition algorithm to improve the recognition accuracy, we have as a null hypothesis that there is no difference between the results of the two algorithms, i.e. $\mu_Z = 0$. Using the value of W, the normal distribution and a one-tailed test, we can infer the P value describing the probability that the improvement in the recognition result is indeed statistically significant. A usual level of significance is 0.05.

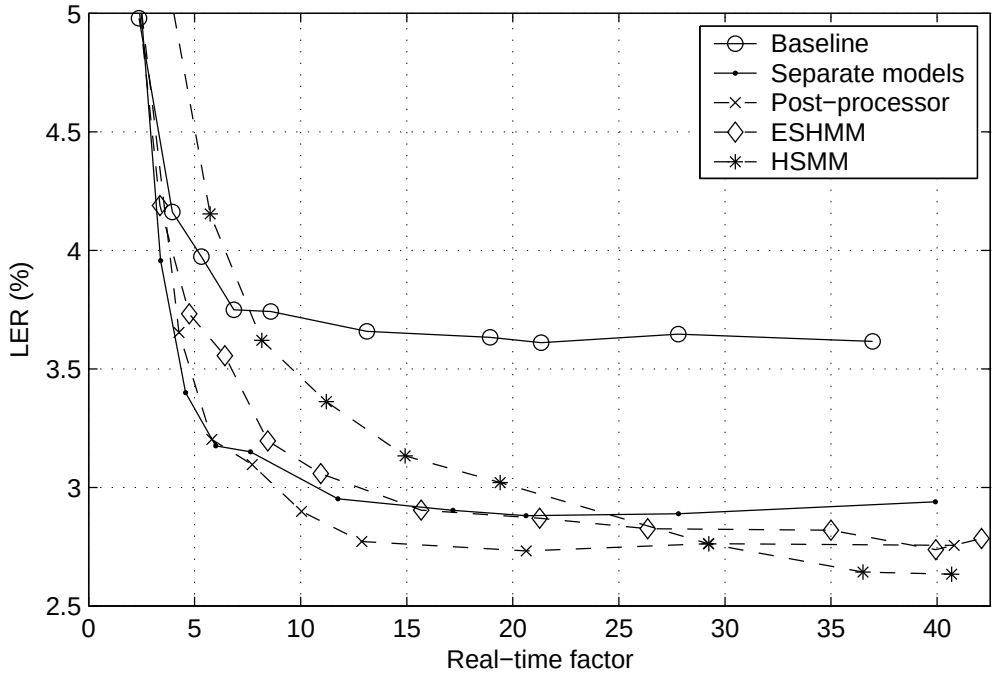


Figure 5.1: Letter error rate as a function of real-time factor for the best variants of the different duration modeling techniques.

For the evaluations performed for this thesis, the about half an hour test material was divided to 25 separate segments with slightly varying durations. This is somewhat too few for the statistical analysis described above to be reliable, as in [12] the use of at least 50 segments is suggested. This is because now the distribution of W may not be close enough to the normal distribution, thus affecting the reliability of the statistical deductions. However, using more segments would reduce the durations of the segments and therefore increase the variance of the accuracy measurements of the individual segments and also degrade the performance of the language model. To get the best possible estimate for the final accuracy measurements, a simple mean over the accuracy measurements of the individual segments was computed.

5.2 Results

All the duration modeling techniques presented in Chapter 4 were implemented to the existing speech recognition system and evaluated with the same test material. Also some variants for the different techniques were evaluated. Figure 5.1 shows the letter error rate of the best variants of each duration modeling technique, plotted against the real-time factor. The markings in the plot represent the different measurement

points in which the recognition test were run. These points were controlled by the beam parameter of the decoder, and therefore no direct control over the exact real-time factor was possible.

The baseline result shown in the Figure 5.1 represents the model in which the single and double phonemes are not separated to their own acoustic models. Therefore the distinction between the two variants are made using only the language model. This was the starting point for this thesis. The first improvement was the model labeled as "separate models", which has separate HMM models for the two phoneme variants. This makes it possible to model the duration variations using the transitions of the three HMM states. It also separates the emission densities of the HMM states of the phoneme variants, although phonetically they should be the same. The emission densities could be shared among different HMM models, but they were still kept separate to keep the models conceptually as simple as possible. Besides, the duration of the phones may affect at least the quality of articulation, and therefore this separation is reasonable. The actual duration modeling techniques were based on the models with separated phonemes, upon which the durations were explicitly modeled.

The plot shows how the real-time factor, and therefore the beam parameter of the decoder, affects the choice for the duration modeling technique. With running speeds with real-time factor less than 8, none of the duration modeling methods improve the recognition accuracy. With real-time factors 8 to 28 the post-processor model seems to be the best. But if the decoder is run with settings where the accuracy of all the methods with respect to the real-time factor has converged, HSMM ends up being the overall best duration modeling technique.

Figure 5.1 also shows that after the recognition accuracy for a certain model has converged, the curve begins to fluctuate and rise slightly. This is due to noise in the recognition results and also because with larger beam values the methods are presented with more morph alternatives among which to choose the best one, which may add some more confusion to the selection. Because the main point in this work was to compare the duration modeling techniques with each other, it is fair to choose the best possible measurement point for each technique for the final accuracy comparison. Next, each of the duration modeling techniques are considered separately, their variants are presented, and the results are compared to the "separate models" model. This reference model can be seen as the direct way of dealing with different phone durations emerging from the single and double phonemes, a difference which is crucial to be dealt with in Finnish.

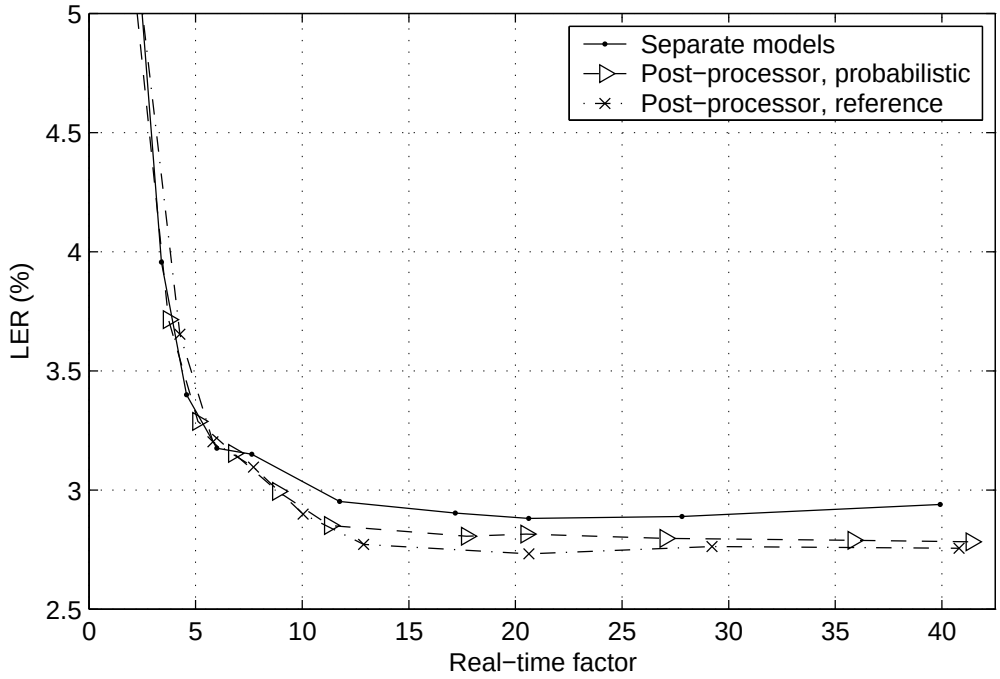


Figure 5.2: Recognition measurements with the post-processor duration model.

Post-processor duration model

In Section 4.5 two derivations of the post-processor duration model with slightly different results were presented. Figure 5.2 shows the performance of these models with respect to the real-time factor. For comparison, the results for the "separate models" model is shown, which has exactly the same acoustic models as used with the two post-processor techniques. The only difference is therefore that the HMM state durations are not modeled explicitly. The state duration statistics were measured at the final training iteration and modeled using gamma distributions.

The figure shows that the more heuristically derived variant of the post-processor model, the one introduced in [17] and therefore called the "reference" model, performs better than the probabilistic variant derived in Section 4.5, shown as the "probabilistic" model. The difference between these is that with the "reference" model, the HMM transition probabilities are included with the final duration probability measure. This suggests that these transition probabilities may carry some additional information which is not available in the gamma distributions of the state durations.

Table 5.1 shows the letter and word error rates of the best recognition results among these models. The real-time factor at which this best result was obtained is shown

Table 5.1: Comparison of the post-processor duration model performances.

Model	RT-factor	LER (%)	WER (%)
Separate models	20.6	2.88	16.2
Post-processor, probabilistic	41.4	2.78	15.5
Post-processor, reference	20.6	2.73	15.3

in the field "RT-factor". The better post-processor model achieves about 5% relative improvement in the letter error rate when compared to the model without explicit duration modeling. For the other post-processor variant, the improvement is 3.5%, and it is achieved with much slower settings.

HSMM

For hidden semi-Markov models, two variants were tested to see the effect of training the models using the improved duration modeling. The untrained model is therefore the same as the "separate models" model, except that the duration distribution statistics measured at the training time were used to explicitly model the state durations in the form of gamma distributions. The trained variant, on the other hand, used the untrained model as a starting point for additional training. Four iterations over the 12 hour training material were run, after which the models had reached convergence. Figure 5.3 shows the results, again as a function of the real-time factor and with the "separate models" model as the reference.

As expected, for HSMM a much slower recognition is required to achieve even the accuracy of the normal HMM model. But as the beam parameter is increased, the accuracy keeps increasing, and becomes finally better than with any other duration modeling technique. In [3] an increase of computational effort of 3.2 times with respect to conventional HMM was reported, when using the same kind of HSMM implementation as in this thesis. However, the assumptions of that measurement were not clearly stated. When considering the Figure 5.3, it is clear that such statements are not unambiguous, because the running time can be adjusted at the expense of the accuracy.

The trained model is constantly better than the untrained one, without training the performance is even slightly worse than with the better post-processor model. The disadvantage with the trained model is that using the HSMM models in training more than doubles the time of the training iterations. But as the training is done prior to the recognition, its speed requirements are not that important. Table 5.2 shows the final best recognition accuracy measurements. The trained HSMM results in about 8% reduction in the letter error rate when compared to the model without

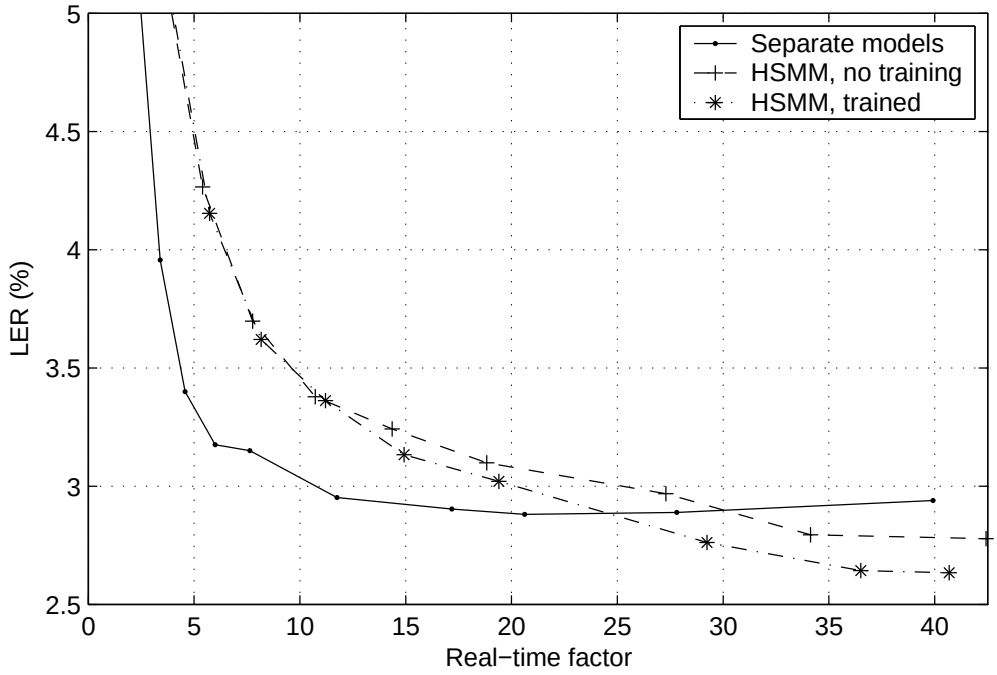


Figure 5.3: Recognition measurements with HSMM duration model.

Table 5.2: Comparison of the HSMM duration model performances.

Model	RT-factor	LER (%)	WER (%)
Separate models	20.6	2.88	16.2
HSMM, no training	42.4	2.78	15.5
HSMM, trained	40.7	2.63	15.2

explicit duration modeling.

ESHMM

With expanded state hidden Markov models the use of different state duration distributions were tested. The other distribution with which the ESHMM was evaluated was the gamma distribution used also with the other duration modeling techniques. In addition to that, a less restricted, non-parametric distribution model was used. This distribution was obtained from the measured duration distribution by smoothing it with a Gaussian window, which width corresponded to the duration of 5 frames. With gamma distribution, the HMM states were expanded on average to 3.8 sub-

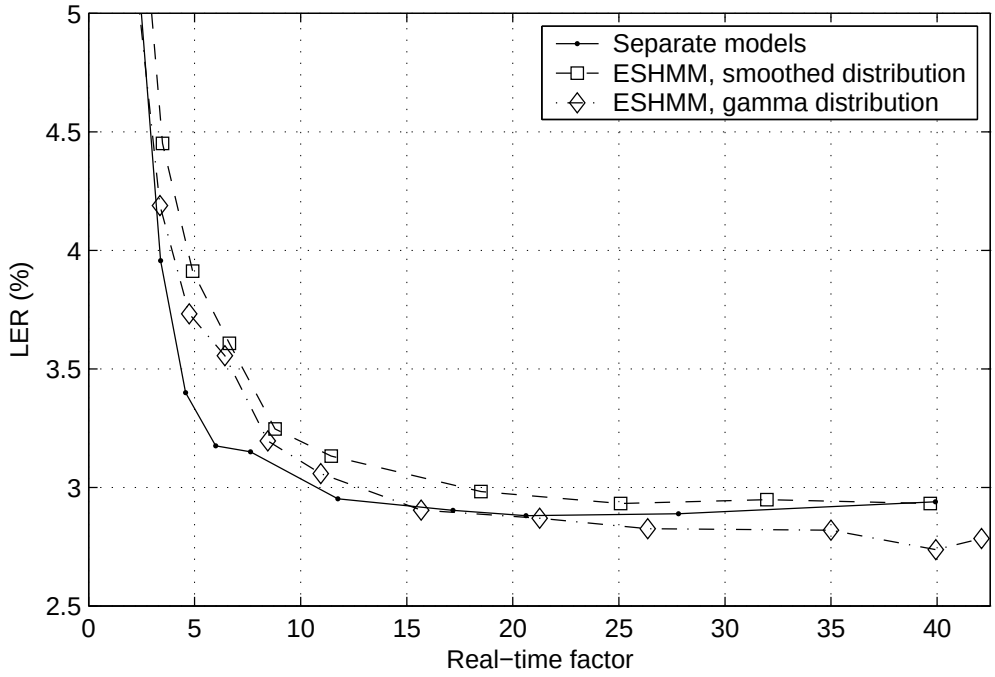


Figure 5.4: Recognition measurements with ESHMM duration model.

Table 5.3: Comparison of the ESHMM duration model performances.

Model	RT-factor	LER (%)	WER (%)
Separate models	20.6	2.88	16.2
ESHMM, smoothed distribution	25.1	2.93	16.7
ESHMM, gamma distribution	39.9	2.74	15.8

HMM states, and with the smoothed distribution on average 3.5 sub-HMM states were used. After the generation of the sub-HMMs, the ESHMM models were trained for six iterations, so that the models achieved convergence. The sub-HMM distributions were kept properly constrained also in the training.

Figure 5.4 shows the recognition results for these two distributions with ESHMM, and also the reference "separate models". The results confirm that gamma distribution is a good distribution for state durations, as it outperforms the non-parametric distribution. Although more smoothing could have improved the performance of the non-parametric distribution, it would also reduce the time resolution of the distribution and therefore might degrade the actual modeling ability. Letter and word error rates with the ESHMM variants are shown in Table 5.3.

The overall performance of the ESHMM was a small disappointment. Despite it was supposed to be more flexible than the post-processor duration model, the ESHMM barely reaches the accuracy of that simple technique. Compared to the HSMM, the ESHMM has in general smaller impact to the efficiency of the recognition, but the best results are still obtained only with very slow decoder settings. One explanation for this might be the pruning strategy of the decoder. Although the functioning of HSMM and ESHMM can be conceptually very similar, they are handled very differently in the decoder.

Statistical significance

As was stated in the beginning of this chapter, it is important to analyze whether the improvements measured in the evaluation really are statistically significant, or whether they could have been caused by the inherent noise in the recognition measurements. Only the two most interesting models are compared here against the "separate models" model. These are the "reference" post-processor, achieving 5% relative improvement in the letter error rate with moderate real-time factors, and the trained HSMM, which achieved 8% relative improvement in LER with rather high real-time factors.

When analyzing the recognition accuracies of independent segments between the post-processor and the "separate models" model, it could be noted that the variance of the LER differences is rather high. The absolute mean LER difference is 0.149% with a variance of 0.457. The post-processor outperformed the comparison model only in 14 out of 25 segments. The LER was the same in three segments and the comparison model was better in 8 segments. When computing the W in Equation 5.1 the result was 1.104. This represents a point in the x-axis of the standardized normal distribution. Therefore the P-value or the probability of the null hypothesis that there is actually no difference between the two methods is $1 - \Phi(1.104)$, where $\Phi(x)$ is the cumulative distribution function of the normed normal distribution. This results in a P-value of 0.135, which is not within the usual significance level of 0.05. The improvements resulting from the application of the post-processor duration model can not therefore be considered as statistically significant.

The difference between the segment accuracies of HSMM and the reference "separate models" has a mean of 0.238% (LER) and a variance of 0.401. The variance still seems to be rather high, and HSMM does not consistently outperform the reference model in each segment either. In 15 segments it was better than the reference model, in two segments the LER was the same and in 8 segments the reference model outperformed the HSMM. Still, from Equation 5.1 we now get $W = 1.878$, leading to a P-value of 0.03. The reduction of the LER resulting from using HSMM can therefore be considered as statistically significant.

The conclusions of the test for statistical significance is that only the improvements gained by the HSMM can be regarded as statistically significant. For the post-processor model, a larger test set would be needed to get reliable measures. On the other hand, as mentioned in the test setup considerations, the assumptions of the matched-pairs test were not completely fulfilled, as the 25 segments over which the W value was computed may not allow the distribution to be assumed as normal.

The variance of the LER differences over the segments depends on the segment durations. The rather high variances suggest that longer segments could make the evaluation more reliable, but as there were no more test material, this would have reduced the number of segments even further. Also what makes the variance of the LER differences seem to be that large is that in one segment there are relatively few errors, and the errors are not independent due to the language model. When comparing the models with and without duration modeling, there might be only a few words in each segment which are recognized differently. The number of letter errors in such a case may depend highly on the language model and the actual words in question. The language model also correlates the recognition of sequential words so that the errors are more likely to appear in bursts.

Chapter 6

Conclusions and Discussion

In this thesis the use of phone duration information as a mean to improve recognition results was studied. Different techniques for incorporating duration models inside the HMM framework of the standard modern speech recognition systems were presented, and some modifications to the existing schemes were derived. Finally the different techniques were compared in experimental evaluations, showing their performance and behavior with different decoder running times.

The work started from a need to improve the existing speech recognition system, which did not discriminate between the single and double phonemes acoustically. The first and the most straightforward modification was therefore to separate these phoneme forms into their own acoustic models. This in fact contributed the largest improvement to the former system, which was shown as the baseline system in the results. The improvements from the actual duration modeling techniques were much smaller, but nevertheless more interesting. After all, with this kind of modern speech recognition system, it is hard to come up with any single method which would improve the recognition accuracy significantly.

In the context of the speech recognizer this study was based on, the post-processor duration modeling technique seems to be the best choice for most purposes. It achieves relatively good improvements to the recognition accuracy with moderate running times. Besides being computationally efficient, it is also very easy to be implemented. It does not affect the time-consuming Viterbi search at the heart of the speech recognizer's decoder, but merely reranks the hypothesis with better duration distributions than the standard HMM framework offers. It does not involve modifications to the training part of the system, others that collecting the duration statistics for the phones, which is easy from the speech segmentation resulting from the model training.

The best results were still obtained by replacing the HMMs with hidden semi-Markov models. However, this method slows down the recognition so that the improvements are only gained if the decoder is run with settings slow enough. Moreover, the implementation of the HSMM to the decoder is significantly harder than for the other techniques, and the best results required implementing the duration models also to the training part of the system. The expanded state hidden Markov model should have been some kind of a compromise between these two methods, as it is easy to be implemented to the normal HMM framework but still holds possibility for modeling the durations very flexibly. However, its performance remained quite poor, possibly due to the pruning strategy of the decoder.

The preferred post-processor duration modeling technique achieved about 5% relative reduction in the letter error rate of the recognizer when compared to the system without explicit duration modeling. If the decoder was run slow enough (real-time factor about 40 in the current, rather unefficient system), the HSMM resulted in 8% relative reduction in the letter error rate. Unfortunately only the latter improvement could be considered as statistically significant. Larger test set would be needed to confirm that also the post-processor technique really improves the recognition performance.

What does these improvements then tell about the use of durations with speech recognition? The statistically significant improvement to the recognition accuracy shows that the poor duration modeling capability of the standard HMMs is not enough, at least for applications similar to the evaluation task of this thesis. However, the usefulness of the described duration modeling techniques directly depends on the quality of the underlying statistical models of phone durations. As the test material was a book spoken by one professional speaker, the speech was steady enough so that the phone durations did not contain excess variance. Therefore it was enough to do direct measurements about the durations without considering the number of factors affecting them, such as the speaking rate, the stress or syllable and word contexts. Only the phoneme context was modeled to the extent of using triphone models. The speaking rate adaptation was discussed in Chapter 4, but not implemented. One reason for this was that incorporating these now ignored sources of phone duration variance into the acoustic models, or even measuring them in some reasonable way, is not at all easy.

With this study it remains unclear whether the recognition accuracy in the evaluation task now performed could be further improved if these variation factors were taken into account. When moving from models designed for one clearly and steadily speaking speaker to more realistic speaker independent real-world situations, it is very probable that these things would begin to dominate over the exact modeling technique. Without a careful examination of at least some of these factors, the improved phone duration modeling ability might be lost in the poor statistical models of the actual duration distributions.

Analyzing the improvements gained by incorporating some new modeling method or technique in an existing speech recognizer is not that straightforward. The state-of-the-art speech recognition system is today well established after a development of over two decades, and the components of such a system have been optimized over time for good mutual operation. Replacing one component with a novel one is therefore likely to break up this collaboration. In [4] it is stated that even an initial increase in the error rate should be tolerated when evaluating new methods, as long as there are solid motivations for using the new methods and that something can be learned by investigating them. With this in mind, it is very promising that real recognition improvements were achieved by using explicit duration modeling, without affecting the rest of the system. But the improvements might be even better if the rest of the system could be adjusted to work well with the new technique. For example, using three HMM states to model each phone or triphone is simply a de facto standard, but when using duration modeling some other number of states might be optimal.

Besides of the three duration modeling techniques studied in this thesis, at least one possibly useful duration modeling family has been presented in the literature. The HSMMs can be implemented in various non-optimal ways, which significantly reduce the impact of the duration model to the recognition efficiency. A prior belief was that a good mathematical model was needed to be able to gain something from the phone durations, and therefore only the optimal HSMM was studied. However, as the evaluations with the post-processor model indicated, also simple models may work reasonably well, and the benefit is that the efficiency is not penalized as much as with the more complex models.

Bibliography

[1] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. A new algorithm for the estimation of hidden Markov model parameters. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 493–496, 1988.

[2] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] Antonio Bonafonte, Xavier Ros, and Jose B. Mariño. An efficient algorithm to find the best state sequence in HSMM. In *Proceedings of Eurospeech*, pages 1547–1550, 1993.

[4] Hervé Bourlard, Hynek Hermansky, and Nelson Morgan. Towards increasing speech recognition error rates. *Speech Communication*, 18(3):205–231, May 1996.

[5] Hervé Bourlard, Yochai Konig, and Nelson Morgan. Remap: Recursive estimation and maximization of a posteriori probabilities. Technical Report TR-94-064, International Computer Science Institute, March 1995.

[6] David Burshtein. Robust parametric modeling of durations in hidden Markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 548–551, 1995.

[7] Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02*, pages 21–30, 2002.

[8] Thomas H. Crystal and Arthur S. House. Characterization and modeling of speech-segment durations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2791–2794, 1986.

[9] Thomas H. Crystal and Arthur S. House. Segmental durations in connected-speech signals: Current results. *Journal of Acoustic Society of America*, 83(4), April 1988.

[10] R. Falthauser, T. Pfau, and G. Ruske. On-line speaking rate estimation using Gaussian mixture models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1355–1358, 2000.

[11] Mark J. F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on speech and audio processing*, 7(3), May 1999.

[12] L. Gillick and S. J. Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 532–535, 1989.

[13] Teemu Hirsimäki. A decoder for large vocabulary continuous speech recognition. Master's thesis, Helsinki University of Technology, 2002.

[14] Michael M. Hochberg and Harvey F. Silverman. Constraining model duration variance in HMM-based connected-speech recognition. In *Proceedings of Eurospeech*, pages 323–326, 1993.

[15] Matti Jalanko. *Studies of Learning Projective Methods in Automatic Speech Recognition*. PhD thesis, Helsinki University of Technology, 1980.

[16] Esther Janse. Word perception in fast speech: artificially time-compressed vs. naturally produced fast speech. *Speech Communication*, 42:155–173, 2004.

[17] B. H. Juang, L. R. Rabiner, S. E. Levinson, and M. M. Sondhi. Recent developments in the application of hidden Markov models to speaker-independent isolated word recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 9–12, 1985.

[18] Shigeru Katagiri, Chin-Hui Lee, and Biing-Hwang Juang. New discriminative training algorithms based on the generalized probabilistic descent method. In *Proceedings of the IEEE Workshop on Neural Networkds for Signal Processing*, pages 299–308, 1991.

[19] Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-35(3):400–401, March 1987.

[20] Teuvo Kohonen. The "neural" phonetic typewriter. *Computer*, 21(3):11–22, March 1988.

[21] Mikko Kurimo. *Using Self-Organizing Maps and Learning Vector Quantization for Mixture Density Hidden Markov Models*. PhD thesis, Helsinki University of Technology, 1997.

[22] Stephen E. Levinson. Continuously variable duration hidden Markov models for speech analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1241–1244, 1986.

[23] John Makhoul, Salim Roucos, and Herbert Gish. Vector quantization in speech coding. *Proceedings of the IEEE*, 73(11):1551–1588, November 1985.

[24] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[25] J. S. Milton and Jesse C. Arnold. *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computer Sciences*. McGraw-Hill, 3rd edition, 1995.

[26] Nelson Morgan, Eric Fosler, and Nikki Mirghafori. Speech recognition using online estimation of speaking rate. In *Proceedings of Eurospeech*, pages 2079–2082, 1997.

[27] A. Noll and H. Ney. Training of phoneme models in a sentence recognition system. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1277–1280, 1987.

[28] Douglas O'Shaughnessy. *Speech Communication: Human and Machine*. Addison-Wesley, 1987.

[29] Joseph W. Picone. Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 81(9), September 1993.

[30] Thomas F. Quatieri. *Discrete-Time Speech Signal Processing: Principles and Practice*. Prentice Hall PTR, 2002.

[31] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), February 1989.

[32] Padma Ramesh and Jay G. Wilpon. Modeling state durations in hidden Markov models for automatic speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 381–384, 1992.

[33] Lennart Råde and Bertil Westergren. *Mathematics Handbook for Science and Engineering*. Studentlitteratur, 4th edition, 1998.

[34] M. J. Russell and R. K. Moore. Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5–8, 1985.

[35] Martin J. Russell and Anneliese E. Cook. Experimental evaluation of duration modelling techniques for automatic speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2376–2379, 1987.

[36] K. Samudravijaya, Sanjeev K. Singh, and P. V. S. Rao. Pre-recognition measures of speaking rate. *Speech Communication*, 24:73–84, 1998.

[37] Richard Schwartz, Steve Austin, Francis Kubala, John Makhoul, Long Nguyen, Paul Placeway, and George Zavaliagkos. New uses for the N-Best sentence hypotheses within the BYBLOS speech recognition system. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–4, 1992.

[38] Matthew A. Siegler and Richard M. Stern. On the effects of speech rate in large vocabulary speech recognition systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 612–615, 1995.

[39] Vesa Siivola, Teemu Hirsimäki, Mathias Creutz, and Mikko Kurimo. Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In *Proceedings of Eurospeech*, pages 2293–2296, 2003.

[40] Vesa Siivola, Mikko Kurimo, and Krista Lagus. Large vocabulary statistical language modeling for continuous speech recognition in Finnish. In *Proceedings of Eurospeech*, pages 737–740, 2001.

[41] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.

[42] Kari Torkkola, Jari Kangas, Pekka Utela, Sami Kaski, Mikko Kokkonen, Mikko Kurimo, and Teuvo Kohonen. Status report of the finnish phonetic typewriter project. In Teuvo Kohonen, Kai Mäkisara, Olli Simula, and Jari Kangas, editors, *Artificial Neural Networks (ICANN-91)*, volume 1, pages 771–776. North-Holland, 1991.

[43] S. V. Vaseghi. State duration modelling in hidden Markov models. *Signal Processing*, 41:31–41, 1995.

[44] Jan P. Verhasselt and Jean-Pierre Martens. A fast and reliable rate of speech detector. In *Proceedings of the International Conference on Spoken Language Processing*, pages 2258–2261, 1996.

[45] Kalevi Wiik. *Finnish and English Vowels*. Turun yliopisto, 1965.

[46] Kalevi Wiik. *Taksonomista fonologiaa*. Painosalama Oy, 1989. (in finnish).

[47] Kalevi Wiik. *Fonetiikan perusteet*. WSOY, 2nd edition, 1998. (in finnish).

[48] Daniel Willett, Christoph Neukirchen, and Gerhard Rigoll. DUcoder-the Duisburg University LVCSR stackdecoder. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1555–1558, 2000.