THESIS ON INFORMATICS AND SYSTEM ENGINEERING C__

Methods for Estonian Large Vocabulary Speech Recognition

TANEL ALUMÄE

Faculty of Information Technology
Department of Informatics
TALLINN UNIVERSITY OF TECHNOLOGY

Dissertation was accepted for the commencement of the degree of Doctor of Philosophy in Engineering on November 1, 2006.

Supervisors: Prof. Emer. Leo Võhandu, Faculty of Information Technology
Einar Meister, Ph.D., Institute of Cybernetics at Tallinn University of Technology

Opponents: Mikko Kurimo, Dr. Tech., Helsinki University of Technology
Heiki-Jaan Kaalep, Ph.D., University of Tartu

Commencement: December 5, 2006

Declaration: Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology has not been submitted for any degree or examination.

/ Tanel Alumäe /

ISSN 1406-4731
ISBN 9985-59-661-7

Contents

Chapter 1

Introduction

The focus of this thesis is on developing efficient models and methods for large vocabulary speech recognition for Estonian. In the introduction, the speech recognition problem is first briefly described. The reasons for investigating language-specific issues are then given together with an overview of most important language-specific aspects of speech recognition. Next, some related work concerning Estonian speech recognition and large vocabulary speech recognition of other similar languages is introduced. The final sections outline the scope and the approaches that are developed throughout the thesis.

1.1 The speech recognition problem

Speech recognition is the process of converting an acoustic signal representing a spoken utterance or a longer speech passage, captured by a microphone or a telephone, to a list of words that is hopefully close to the original word sequence.

Speech recognition systems can be characterized by many parameters, such as speaking mode, speaking style, speaker independence, vocabulary size, language model, usage environment and input channel. An isolated-word speech recognition system requires that the speaker pause briefly between words, whereas a continuous speech recognition system does not. Speaker independent systems can be used without speaker enrollment while speaker-dependent systems require a transcribed speech sample of a user's speech to adapt the system to his or her voice and speaking style. System's vocabulary defines the words that the system knows about. Vocabulary size is considered small if the number of words is below 100, and large if the number of words is more than 20 000. System's language model defines the different combinations in which the words can be combined in a sentence and may also estimate the likelihood of different word sequences. The simplest language model can be specified as a finite-state network, where the permissible words following each word are given explicitly. More general

language models approximating natural language are specified in terms of a statistical context-sensitive models.

This thesis focuses on speaker-independent, large vocabulary continuous speech recognition (LVCSR). There are many application areas for LVCSR technology. The most obvious of them is desktop dictation, where user speaks into the microphone and the computer automatically converts it to textual representation. Desktop dictation has some advantages over typing the text using a keyboard: using spoken language, it is easy to achieve a data input rate of 150-250 words per minute [Schukat-Talamazzini, 1995, p. 1]; in order to achieve this, no long-lasting training is required; additionally, spoken language interface leaves user's hands and eyes free for other activities, and grants more freedom of movement.

In addition to desktop dictation, there is a growing need for more "industrial" usages of LVCSR. There exist huge amounts of archived untranscribed speech data, e.g. radio and television broadcast archives, recorded meetings, lectures, speeches and debates. Currently, the only way to find certain excerpts from such archives or analyse the content of them is to rely on some available metadata of the recordings or just listen to them. The advance of LVCSR would make it possible to automatically transcribe the speech in the archives and make them readable and accessible for automatic retrieval. Also, there are many areas where spoken language must be always transcribed. In such cases, currently human transcribers create verbatim transcripts for speeches, conversations, legal proceedings, meetings, and other events when written accounts of spoken words are necessary for correspondence, records, or legal proof. With the improvement of speech recognition technology, some of such laborious manual work could be replaced with an automatic transcription system, or human transcribers could be assisted by a system which automatically prepares a draft version of the transcription.

A growingly important field where LVCSR technology plays an significant role is automatic speech-to-speech translation. This scenario requires both speech recognition and speech synthesis technology of all participating languages. In addition, sophisticated multilingual spoken language understanding is needed. Speech-to-speech translation is especially relevant in Europe where a wide variety of languages is spoken and there is a strong need for interlingual communication, for example in the context of European Union institutions.

The general problem of automatic recognition of speech by any speaker in any environment is still far from being solved. But in the last decades, there have been great advances in the area of LVCSR. For languages with a large number of speakers, such as English, French and German, many successful speech recognition systems have been developed and commercial systems are widely available. For smaller languages like Estonian, there is little interest from commercial vendors to develop such technology. However, according to a futuristic view, it is highly important for the survival of small languages to develop

human language technologies for the language, including tools for both written and spoken language processing.

The most widely used and successful approach to modern speech recognition is based on statistical and data-driven methods. In this case, no explicit knowledge about the language is programmed into the system. Instead, speech is modelled using well-defined statistical algorithms that can automatically extract knowledge from large amounts of training data. During system development, tens or hundreds of hours of transcribed speech from various speakers are used to train acoustic models. Given such data, statistical techniques can automatically align all speech with the given transcripts and derive the qualitative and temporal aspects of different basic speech sounds in various contexts. On the other hand, large text corpora, possibly containing millions of words, are used to automatically learn the words that are used in the language, and the contexts in which they typically occur. The two trained knowledge sources, or models, together with a pronunciation lexicon that maps all words in the language to sequence of basic speech sounds, can be used during recognition to convert speech into string of words. Such statistical approach is also adapted in this study.

1.2 Language specific aspects of speech recognition

The general architecture of a large vocabulary speech recognition system is language independent. For the large majority of languages, an efficient system can be built using the same kind of components, including a feature extraction front end, hidden Markov model (HMM) based acoustic models (see section 2.3.1), a pronunciation lexicon, a statistical language model and a decoder. However, some details of the design of some of those components can have language specific aspects. The following are the most important design issues that have to be considered when developing a large vocabulary recognition system:

- *Amount and quality of training resources*: the importance of large corpora for training acoustic and language models cannot be overemphasized. Language-specific training data are needed for building robust models for use in a recognition system. This includes both transcribed speech corpus for training acoustic models as well as text resources for estimating statistical language models. To build a good quality recognition system with a medium-sized (10 000-20 000 words) vocabulary, at least 10 hours of transcribed speech material is needed [Lamel et al., 1996].

- *Selection of features to be extracted from speech*: usually, standard Mel frequency cepstral coefficients (MFCC, see section 2.2.2) or perceptual linear prediction (PLP) based features are used, but for some languages (e.g. tone languages, such as Mandarin), other methods are sometimes used in addition to or instead of the standard feature extraction techniques.

- *Choice of basic units for acoustic modelling*: most modern recognizers use units that directly correspond to phonemic inventory of the target language. However, this is not a straightforward decision: first, it might not be clear, which phonemes actually exist in the language. For example, in many languages, dipthongs such as in *day* or *my* in English are considered different phonemes, while in Estonian they are described as a sequence of two qualitatively different vowel phonemes [Eek and Meister, 1999]. The same applies for geminates (long consonants), and to the handling of short and long phonemes. In addition, there are other aspects that may have to be considered (such as tone and stress) when selecting the appropriate units for acoustic modelling for a language. In addition, for some languages, units that are longer (e.g. syllable) or shorter than a phoneme might be more appropriate.

- *Vocabulary selection and language modelling*: the vocabulary of a recognizer defines the words that can be recognized and the language model defines their prior probabilities in various contexts. A typical method for selecting vocabulary is to choose the top 20 000-60 000 most frequent words in the training text corpus. While this works well for languages like English, it is not suitable for highly inflective and compounding languages, since each inflected word form is considered as a different word. As a consequence, the number of different words is very large, and a high out-of-vocabulary rate is expected when only 60 000 most frequent words can be recognized. Increasing the vocabulary size may solve the out-of-vocabulary problem but it cannot reduce the severeness of another issue caused by the large number of different words – data sparsity. Due to the high number of different words, many of them are only rarely seen in the training corpus which makes it difficult to robustly estimate their prior probabilities in various contexts. Instead, for highly inflective languages, usually some type of sub-word units are used as basic units in a language model which are recombined into words after decoding. The actual method for statistical language modelling also depends on other features of the language, available linguistic and other processing tools, and the size of the available text corpus.

- *Pronunciation modelling*: when appropriate sub-word acoustic models are used, a correct pronunciation for each word must be defined so that concatenation of basic acoustic units can accurately represent the word to be recognized. The mapping is based on language-specific knowledge. In some languages, simple grapheme-to-phoneme conversion can be used for determining the pronunciation for each word. Other languages need large standard or hand-crafted pronunciation lexicons, often combined with knowledge-based or data-driven conversion rules. For some words, such as *tomato* in English, we might need to provide alternative pronunciations. In

addition, various continuous speech phenomena, such as reduced sounds and assimilation have to be considered, in order to make pronunciation lexicon more robust.

In an analysis by Bell Laboratories, the time for developing a large vocabulary speech recognition system for a new language is estimated to be around 9-15 work months [Gokcen and Gokcen, 1997] and cost several hundred thousand dollars. This does not include collecting of the speech and text material needed for training which could easily be more expensive that the development itself.

1.3 Related work

In spite of active research in the area of phonetics and computational linguistics, the research in the area of speech recognition for the Estonian language has been not very active. However, the first research results about acoustic analysis of Estonian vowel and consonant system and prosody date back to 1960s [Lehiste, 1966].

In the end of 1980s, experiments with recognizing words differing in distinctive quantity (e.g. *kade-kate-katte*) were made, using spectral match and dynamic programming techniques and including probabilities of state durations and state duration ratios as an additional factor in determinining the best path [Kuhn and Ojamaa, 1989]. The authors conclude that distinguishing words differing only in distinctive quantity is a major problem with varying speech rates and it could not be completely overcome, even when using likelihoods of state duration ratios.

In the 1980s, some experiments on vowel recognition using electronic filterbanks were carried out by E. Künnap [Künnap, 1992].

During the last decade, the research on spoken language technology in Estonia has been carried out mainly at the Laboratory of Phonetics and Speech Technology, Institute of Cybernetics at Tallinn University of Technology. In 1995-96, neural nets were used for diphone recognition experiments [Meister, 2001]. The preliminary tests reached a classification rate of about 70% for all diphones. In 2000, a prototype for isolated word recognition (Estonian numbers and names of Estonian letters) was developed in co-operation with Institute of Engineering Cybernetics of Minsk [Meister et al., 2001]. The system used continuous dynamic time warping techniques for word recognition. The recognition system was used for developing a spoken dialogue system for parking system over mobile phone. A speaker-independent recognition rate of 72 for numbers as 58 for letters was achieved. Dialogue success rate was 92%, when two repetitions to fix misrecognized words were accepted.

As part of the author's masters thesis, a limited vocabulary connected speech recognition system based on hidden Markov models as context-dependent phone units was developed [Alumäe, 2002, Alumäe, 2003, Alumäe and Võhandu, 2003,

Alumäe and Võhandu, 2004]. The aim of the work was to develop a vocabulary-independant basis for Estonian speech recognition. A prototype system for number recognition reached high accuracy. However, no attempts in statistical language modelling and large vocabulary recognition were made.

The work on large vocabulary recognition began with the start of author's doctoral studies. First, a general framework of morpheme-based language model was developed and its performance was compared with a word-based language model [Alumäe and Võhandu, 2004]. The performance of the morpheme-based model was improved using a statistically derived class model [Alumäe, 2004a, Alumäe, 2004b]. Some phonological and morphological modelling issues were investigated in [Alumäe, 2005a]. The two-pass approach using morphological analysis and factored language models in the second pass was introduced in [Alumäe, 2005b] and refined in [Alumäe, 2006].

The recognition problem of other similar highly inflecting and compounding languages (e.g. Finnish, Hungarian, Turkish) has been extensively studied. Many methods have been proposed to deal with the problem of vocabulary growth in large vocabulary speech recognition. Most of the approaches split the words in the vocabulary of a language model into smaller units, in order to increase lexicon coverage. In [Hirsimäki et al., 2006], a language-independent algorithm for discovering word fragments in an unsupervised manner from text is proposed. The algorithm uses the Minimum Description Length principle to find an inventory of word fragments that is compact but models the training text effectively. The same approach has been successfully applied also for Turkish and Estonian [Kurimo et al., 2006]. The Estonian LVCSR experiment based on the SpeechDat speech database achieved a word error rate of 47.6%. Other approaches [Szarvas and Furui, 2003, Kwon and Park, 2003] use language-specific morphological analyser to split words into morphemes. More extensive overview of related work in the area of language modelling for highly inflected languages is given in section 5.1.

1.4 Scope of the thesis

This thesis presents methods for building a large vocabulary continuous speech recognition system for the Estonian language. The approach adapts modern general-purpose statistical framework using hidden Markov models for acoustic models, Mel-frequency ceptrum coefficients for acoustic features and statistical N-grams for language models.

The thesis concentrates on the language specific design issues of the three pre-built knowledge sources that are applied during recognition: the acoustic model, the language model and the pronunciation lexicon. The research on acoustic models tries to find a suitable inventory of basic units that could be used for constructing Estonian word models. The work on language model attempts

to overcome Estonian specific problems in building an efficient statistical N-gram model for large vocabulary recognition. Some issues concerning complex language syntax and training data sparsity are addressed. An algorithm for generating pronunciation lexicon that maps words to sequences of acoustic units is developed. The performance of all developed methods is investigated using a large variety of experiments.

The work relies on the most prominent modern approach to speech recognition problem and thus does not attempt to provide entirely new approaches in the area of feature extraction, acoustic modelling, hidden Markov models and search algorithms.

1.5 Outline of the thesis

Chapter 2 provides a theoretical background of speech recognition in a statistical framework. An introduction to the basic methods in feature extraction, acoustic modelling and language modelling is given. Hidden Markov models are briefly introduced. Chapter 3 introduces properties of the Estonian language. Estonian phonology is described and the phenomena of three distinctive quantity degrees is reviewed. The chapter also presents a brief overview of language orthography, morphology and syntax. In chapter 4, design of Estonian acoustic models for large vocabulary recognition is presented. This chapter also proposes a simple grapheme to phoneme algorithm for generating pronunciation lexicon for Estonian words. Chapter 5 investigates language modelling issues for Estonian large vocabulary recognition. It proposes the use of pseudo-morphemes as basic units for language modelling and describes a method for selecting pseudo-morpheme vocabulary. A statistical method for dealing with compound word reconstruction is proposed. Additionally, the chapter derives two independent methods that attempt to make language modelling more robust. Chapter 6 provides an extensive evaluation of the methods proposed in the previous two chapters. The thesis concludes by discussing the effectiveness of the proposed and evaluated methods. Suggestions for future research are also given.

1.6 Acknowledgements

First and foremost, I thank my supervisors Leo Võhandu and Einar Meister for their guidance throughout my time as a PhD student in Tallinn University of Technology. Their insight, expert advice and enthusiasm were invaluable.

I am grateful to the Laboratory of Phonetics and Speech Technology at the Institute of Cybernetics at Tallinn University of Technology for providing the facilitites and a positive atmosphere for research. I would like to thank my colleague Lya Meister for being kind and always helpful.

I thank the research group of computer linguistics at University of Tartu for providing the text corpora, and people at OÜ Filosoft for providing the morphological analysis software. Those resources have made my work much easier.

A special thanks goes to Dr Elmar Nöth who introduced me to the world of speech recognition at the Friedrich-Alexander University Erlangen-Nuremberg.

I also thank Prof Sadaoki Furui and Takahiro Shinozaki from Tokyo Institute of Technology for their advice and assistance with the Julius decoder.

I thank my long-time employer Aqris Software and especially Oliver Wihler for being supportive to my studies.

I must acknowledge Estonian Information Technology Foundation (EITSA), Estonian Information Technology and Telecommunications Association (ITL) and Estonian Scientific Foundation (ESF, grant 5918) for their financial support.

And finally I thank my family and friends for their patience and support.

Chapter 2

Basic concepts of speech recognition

This chapter describes basic concepts of speech recognition in a statistical framework. Most of modern speech recognition systems can be divided into five basic blocks: the feature extractor, acoustic models, language model, lexicon and the decoder. Figure 2.1 outlines the basic stages of speech recognition.

First, speech signal is digitized and processed by the feature extractor component that transforms the signal into a sequence of feature vectors which are meant to capture the information relevant to the distinction between different speech sounds. In the next step, the feature vectors are decoded, that is, the most likely word sequence hypothesis given the features vector sequence is found. The decoder uses three pre-trained knowledge sources in this process: (1) the acoustic models that model the qualitative and temporal variances of different speech sounds (typically phonemes); (2) the language model that determines what constitutes a possible word, what words are likely to co-occur, and in what sequence; (3) the pronunciation lexicon that maps the words in the language model to a sequence of acoustic units defined by the acoustic model. The decoder uses two sources of information in finding the most likely sentence: the measure of the quality of match between the input features and the valid word sequences, and the probability of hearing a sentence in the language without referring to any acoustic information.

These components of the speech recognizer are reviewed in more detail in the following sections. First, the statistical approach to the speech recognition problem is introduced. Next, components of feature extraction front end are reviewed. The main concepts of hidden Markov models (HMMs) are then presented. Acoustic modelling techniques using HMMs are described. Finally, efficient statistical language modelling techniques and evaluation metrics for language models are presented.

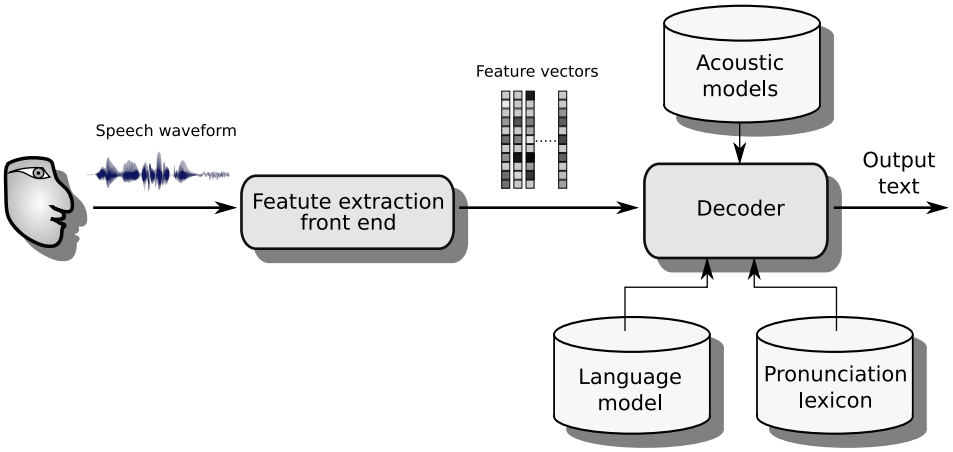

Figure 2.1: Overview of recognition system.

2.1 Probabilistic decoding problem

The speech recognition problem can be described as a decoding problem in communication theory [Shannon, 1948]. The spoken string of words $w = w_1, ..., w_m$ of unknown identity is viewed as passing through an acoustic channel that encodes the words into observable feature vectors $X = x_1, ..., x_T$. The decoder tries to convert the feature vectors X from a coded form back into the original form, i.e. find the most likely word sequence $w^\star$ given X. If the prior probabilities of all possible words sequences $P(w)$ is known, and we also know the conditional distribution of acoustics given words $P(X|w)$, the decoding process can be expanded using the Bayes rule:

$$w^\star = \arg\max_w P(w|X) = \arg\max_w \frac{P(w) \cdot P(X|w)}{P(X)} = \arg\max_w P(w) \cdot P(X|w)$$

The prior probability of different word sequences $P(w)$ can be estimated given a large enough text corpus, and the conditional probability $P(X|w)$ can be estimated given a large corpus of annotated speech, i.e. enough samples of encodings X from word w.

2.2 Feature extraction

The role of feature extraction in the speech recognition framework is to reduce data rate, extract features that are important for subsequent acoustic matching and remove data that is not useful for speech recognition (such as noise, features that a specific to speaker and environment).

2.2.1 Signal acquisition

Before any processing can take place, the speech signal must be acquired and digitized. The digitization is typically done using at least 16 kHz sampling rate and 16-bit A/D conversion which is sufficient for the speech bandwidth (around 8 kHz). However, the open telephone channel is limited to frequency band of 300 Hz - 3.4 kHz. Due to this, 8 kHz sampling rate is typically used when the signal is acquired over the telephone line.

In practice, if the signal is not band-limited, it should be possible to get around 10% relative word error rate reduction when using a sampling rate of 11 kHz instead of 8kHz, and going from 11 kHz to 16 kHz offers further 10% improvement. Further increasing the sampling rate does not have any additional impact to the error rate [Huang et al., 2001, p. 422].

2.2.2 Short-term analysis

Using the raw sampled acoustic waveform of the speech signal for decoding is not practical: if the signal is sampled at 16 kHz using 16-bit accuracy, the amount of data to be processed in each second is 32 kB. Furthermore, the raw signal contains many aspects that are characteristic to the speaker, environment, all of which are not important for speech recognition and are regarded as noise. To reduce the data and extract the important characteristics from the signal, short-time spectrum analysis is used. The most common parameterization is the Mel-frequency cepstral coefficients (MFCC). Using MFCC, speech is transformed to a sequence of typically 39-dimensional feature vectors. The rate of the vectors is commonly 100 per second. At 3900 values per second, this is a large reduction of the original data rate of 32 KB per second.

Feature extraction usually begins by pre-emphasizing the audio to remove glottal and lip radiation effects. The pre-emphasis is implemented by processing the signal using a first order Finite Impulse Filter (FIR) given by

$$y[n] = x[n] - 0.97x[n-1]$$

where $x[n]$ represents the input signal and $y[n]$ the filtered signal. Such filter slightly boost the high frequencies and attenuates the low frequencies.

Next, the pre-emphasized signal is divided into short frames with period of typically 10 ms. In each frame period, signal from a sliding window of 20 or 25 ms is taken for further independent analysis. This process is illustrated in Figure 2.2.

For each analysis frame, a window function such as Hamming is applied first to reduce boundary effects. The window function modifies the input signal $f(n)$ to

$$f^{(m)}(n) = f(n) \cdot w(m-n)$$

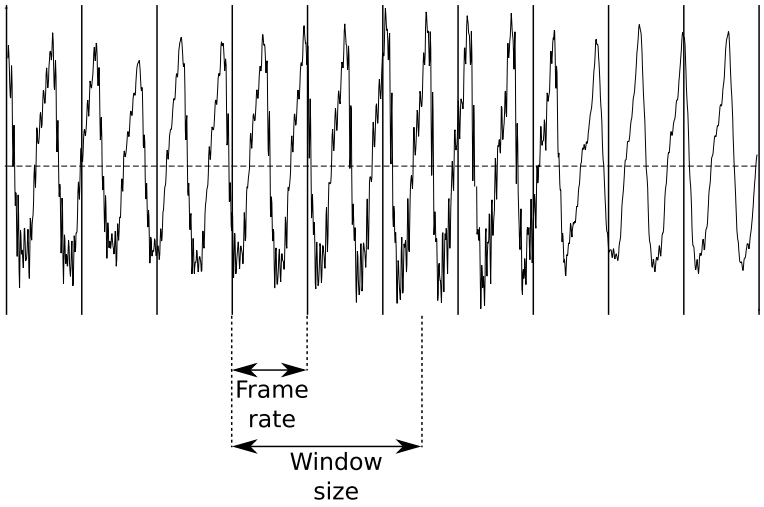

Figure 2.2: Dividing signal into overlapping windows (a frame rate of 10 ms and a window size of 25 ms is used here).

where $w(n)$ is a window function, for instance a Hamming window which is given by

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$$

Constant N is the window size.

Next, a magnitude spectrum of the windowed waveform is computed for each frame using the Discrete Fourier Transform (DFT) which for input signal $f(n)$ is defined as

$$F[n] = \frac{1}{N}\sum_{k=0}^{N-1} f(n)e^{\frac{-jk2\pi}{N}}, n = 0..N-1$$

The linear frequency axis is then warped onto the Mel scale in order to take into account the relationship between frequency and "perceived" pitch. The mapping between the linear frequency scale and Mel scale is given by

$$B(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

The Mel scale is plotted in Figure 2.3.

Next, a bank of partially overlapping triangular filters is taken which compute the average spectrum around each center frequency. The center frequencies are chosen so that they are uniformly spaced on the Mel frequency scale. If f_1 and f_h is the lowest and highest frequency of the filterbank in Hz, F_s is the sampling frequency, M the number of filters (typically 20) and N the size of the DFT, the

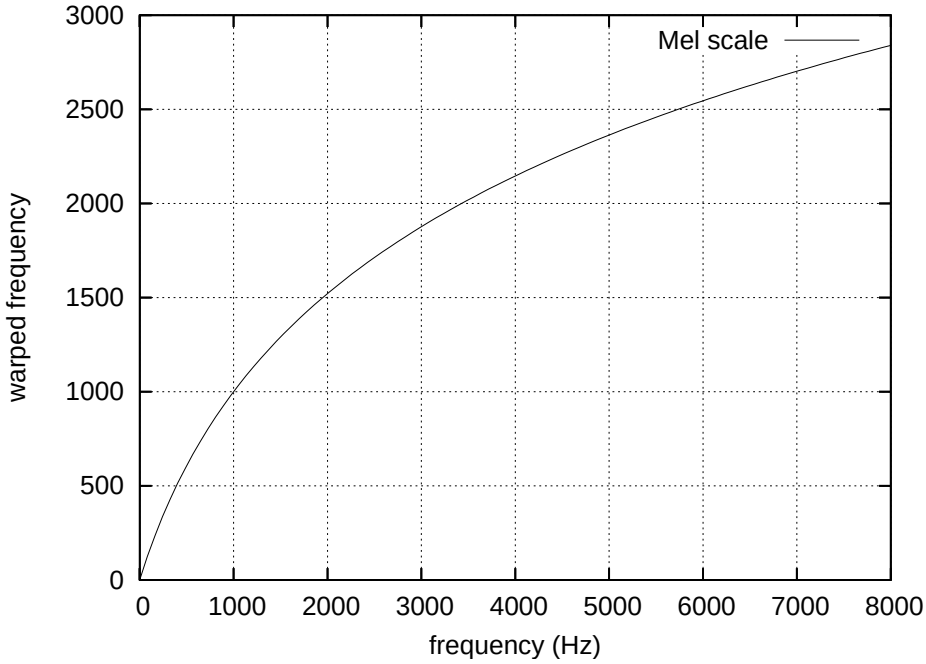

Figure 2.3: Frequency warping according to the Mel scale.

boundary points $f[m]$ can be found by

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_1) + m\frac{B(h_h) + B(f_1)}{M+1}\right)$$

where $B(f)$ is the transformation to Mel scale and B^{-1} its inverse

$$B^{-1}(f) = 700(\exp(f/2595) - 1)$$

Let $H_m[k]$ represent the weight of the jth filter to the kth DFT coefficient and let $|F_{mel}[k]|$ represent the DFT magnitude spectrum warped onto the Mel scale. Then, the filter outputs generate a discrete set of M log-energy terms $e[j], j = 1..M$ which are found by

$$e[j] = \ln\left(\sum_{k=0}^{N-1} |F_{mel}[k]| \cdot H_m[k]\right), j = 1, 2, ..., M$$

Finally, the first 12 Mel-frequency cepstrum coefficients (not including the 0th one) $c_t[i], i = 1..12$ are computed by applying the discrete cosine transform on

the M filter outputs:

$$c[i] = \sqrt{\frac{2}{M}} \sum_{j=1}^{M} \left(e[j] \cos \left(\frac{\pi i}{M} (j - 0.5)) \right) \right)$$

An additional optional step subtracts the cepstral mean $\overline{c}$ from each coefficient $c[i]$, in order to compensate for convolutional distortons, such as different microphone transform functions depending on the speaker distance to the microphone and the room acoustics. Given T coefficients, the cepstral mean is calculated by

$$\overline{c} = \frac{1}{T} \sum_{t=0}^{T-1} c[t]$$

and the normalized coefficients $\hat{c}_t$ by

$$\hat{c}_t = c[t] - \overline{c}$$

The 12 MFCC coefficients are augmented with a normalized log-energy component which is calculated by taking the log of the sum of squared data samples:

$$\tilde{e}_t = \ln \left(\sum_{n=1}^{N_s} s_t^2(n) \right)$$

To capture temporal changes in the spectra, dynamic features are used [Furui, 1986]. Dynamic features measure the change of coefficients over time. Temporal information is particularly useful when using HMMs in acoustic modelling since HMMs assume that each frame is independent of the past. Dynamic features typically consist of first and second order derivatives of the corresponding 13 main features, where the first order derivatives Δc_k for frame k are calculated as

$$\Delta c_k = c_{k+2} - c_{k-2}$$

and second order features $\Delta\Delta c_k$ as

$$\Delta\Delta c_k = \Delta c_{k+1} - \Delta c_{k-1}$$

The dynamic features are appended to the main features, making the final feature vector 39-dimensional.

2.3 Acoustic modelling

Speech is a complex phenomena with a wide qualitative and temporal variability. The variability is caused by coarticulation and contextual effects, physical and

social characteristics of the speaker, speaking style and the properties of the environment and the input channel. Thus, for speech recognition, we need techniques that can adequately model such variability.

Currently, almost all speech recognition systems model sound units by a sequence of connected Hidden Markov Model (HMM) states. These include the CMU Sphinx 2, 3 and 4 [Lamere et al., 2003], Cambridge HTK system [Kim et al., 2005], IBM [Ramabhadran et al., 2006], LIMSI [Lamel et al., 2006], and SONIC [Pellom, 2001] among many others.

In this section, the generative model of HMM is presented. The maximum likelihood (ML) parameter estimation algorithm, the Viterbi decoding algorithm and the Baum-Welch algorithm is briefly described.

2.3.1 Hidden Markov models

In HMM-based speech recognition, it is assumed that the observed sequence of p-dimensional feature vectors is generated by a Markov model as shown in Figure 2.4. A hidden Markov model is a double-embedded stochastic process with the an underlying stochastic process (the state sequence) not directly viewable (thus, the notion of *hidden* Markov models).

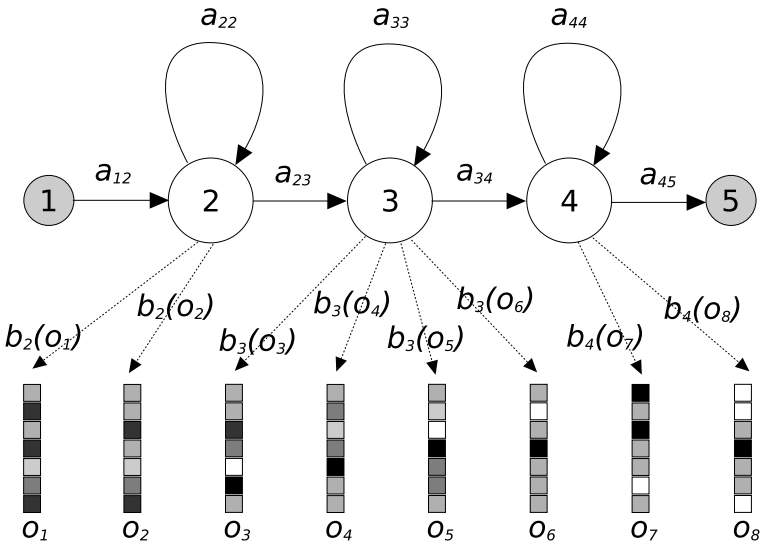

Figure 2.4: Structure of a hidden Markov model.

A hidden Markov model is basically a Markov chain where the output feature vector is a random variable o generated according to the output probabilistic function associated with each state. Formally, it is defined by:

- V – the output observation alphabet. The observation symbols correspond to the physical output of the system being modelled an may be either

discrete or continuous. In speech recognition, the observation alphabet corresponds to the space of possible feature vectors.

- $O = \{o_1, o_2, ..., o_M\}, \;\; o_m \in V$ – an output observation sequence.
- $S = \{S_1, S_2, ..., S_N\}$ – the set of N HMM states.
- $q = q_1 q_2 ... q_T, \;\; q_t \in S$ – the discrete state sequence of the HMM that generates the observation sequence.
- $A = \{a_{ij}\}$ – a transition probability matrix, where $\{a_{ij}\}$ is a probability of taking a transition from state i to state j, i.e.

$$a_{ij} = P(s_t = j | s_{t-1} = i)$$

- $B = \{b_i(o)\}$ – an output probability matrix, where $b_i(o)$ is the probability density function of emitting output observation o when state S_i is entered:

$$b_i(v) = P(o | q_t = S_i), i = 1...N, v \in V$$

  The probability $b_i(o)$ is assumed to be independent of t.

- $\pi = \{\pi_i\}$ – an initial state distribution where

$$\pi_i = P(q_1 = S_i), i = 1..N$$

In speech recognition, it is convenient to extend the basic HMM structure to include initial and final non-emitting states S_0 and S_{N+1}, as shown in the figure. Thus, we can get rid of the initial state probability vector π and incorporate initial state probabilities into state transition vector A.

The state conditional observation vector densities may assume many different forms. The typical choice is a multivariate Gaussian distribution. Its density function is given by

$$b_j(o_t) = \mathcal{N}; \mu, W) = \frac{1}{\sqrt{(2\pi)^{D/2}|W|^{1/2}}} e^{-\frac{1}{2}(o-\mu)^T W_i^{-1}(o-\mu)}$$

where μ and W are the mean vector and the covariance matrix respectively of the distribution and D is the dimensionality of the observation vectors. To reduce the number of free parameters it is usually assumed that the components of the feature vector are uncorrelated, i.e. the off-diagonal elements in the covariance matrix are set to zero. Unfortunately, the "true" parameter vector distributions have often complex shapes and in such case, a single Gaussian density may prove inadequate. This is especially true for a speaker independent system trained on both male and female data. In order to obtain more accurate approximations, it is

common to use mixtures of Gaussian densities

$$b_j(o) = \sum_{m=1}^{M} c_{j,m}\mathcal{N}(o; \mu_{j,m}; W_{j,m}) = \sum_{m=1}^{M} c_{j,m} b_{j,m}(o)$$

where M is a number of mixture components and $c_{j,m}$ is the mixture weight for the mth mixture component in state j. A sample of a one-dimensional probability density function with three Gaussian mixtures is shown in Figure 2.5.

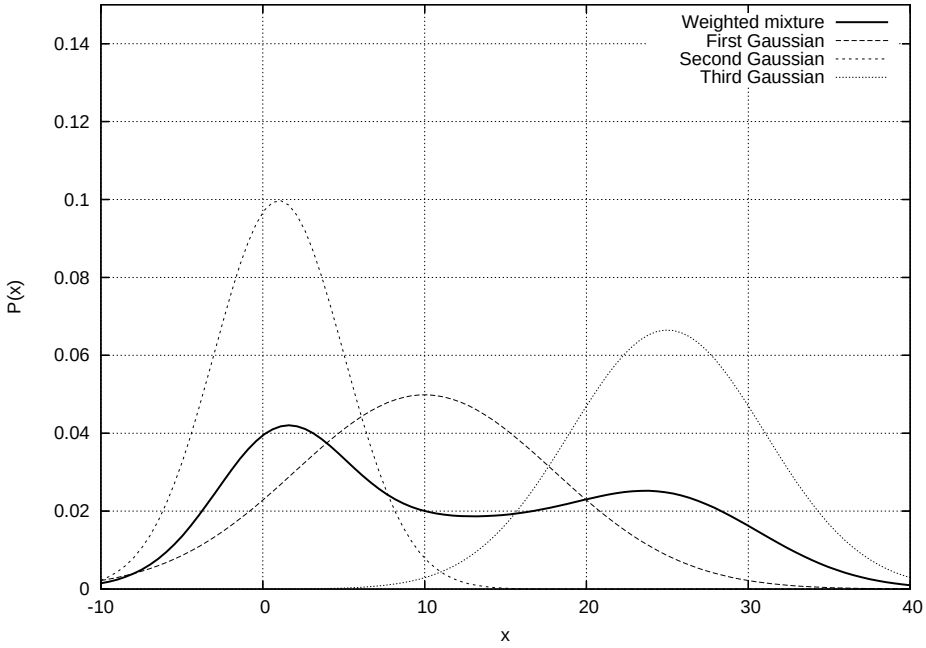

Figure 2.5: One-dimensional probability density function with three Gaussian mixtures.

Given the definition of HMMs, three basic problems must be addressed before they can be applied to real-life applications:

- The evaluation problem – given a model λ and a sequence of observations O, what is the probability $P(O|\lambda)$, i.e. the probability that this model generated the observations?
- The decoding problem – given a model λ and a sequence of observations O, what is the most likely state sequence Q in the model, given that the model generated the sequence?
- The learning problem – given a model λ and a set of observation sequences O, how can we adjust the parameters of the model so that the joint probability (likelihood) $\prod_O P(o|\lambda)$ is maximized?

The evaluation problem

The evaluation problem can be solved with the so-called *forward* algorithm.

If the actual state sequence $q = q_1 q_2 ... q_T$ is known, the probability $P(o|\lambda, q)$ can be found by multiplying the transition and output probabilities that are encountered along the path:

$$P(o|\lambda, q) = \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1},q_t} b_{q_t}(o_t)$$

In reality, the state sequence that generated the observation sequence O is known, and the probability that it was generated by any state sequence can be found by simply enumerating all possible state sequences S of length T and then summing all the corresponding conditional probabilities:

$$P(o|\lambda) = \sum_{q} P(o|\lambda, q) = \sum_{q}^{p} i_{q_1} b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1},q_t} b_{q_t}(o_t)$$

To efficiently solve this equation, the forward algorithm applies dynamic programming techniques to drastically reduce the amount of computation by avoiding the enumeration of paths that cannot possibly be optimal. It uses helper values $\alpha_t(j) = P(o_1 ... o_t, q_t = S_j|\lambda)$ that are defined recursively as:

$$\alpha_t(j) = \begin{cases} \pi_j b_i(o_t) & \text{if } t = 1 \\ \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t) & \text{if } t > 1 \end{cases}$$

This way, the probability under interest $P(O|\lambda)$ can be calculated as

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_T(j)$$

The decoding problem

In speech recognition, it is desirable to find the state sequence $q^\star$ of the model λ that most probably produced the observation sequence O. This problem can be solved using a variant of the forward algorithm, known as *Viterbi* algorithm.

The aim is to find:

$$q^\star = \arg\max_{q \in S^T} P(q|O, \lambda) = \arg\max_{q \in S^T} \frac{P(O, q|\lambda)}{P(O|\lambda} = \arg\max_{q \in S^T} P(O, q|\lambda)$$

Here, instead of the probability $\alpha_t(j)$, we find the maximum of the probabilities $\theta_t(j)$:

$$\theta_t(j) = \max_{q \in S^T} P(o_1 ... o_t, q_t = S_j | \lambda)$$

and the recursive definition becomes:

$$\theta_t(j) = \begin{cases} \pi_j b_i(o_t) & \text{if } t = 1 \\ \max_{i=1}^{N} \theta_{t-1}(i) a_{ij} b_j(o_t) & \text{if } t > 1 \end{cases}$$

The state sequence that maximizes the probability of the observation sequence corresponds to the states that are encountered along the path when calculating

$$\max_{i=1}^{N} \theta_T(j) = P^\star(O|\lambda) := P(O, q^\star|\lambda)$$

The optimization problem

The goal of the optimization problem is to estimate model parameters $\lambda = (A, B, \pi)$ so as to maximize the probability that an observation sequences O where produced by this model:

$$\lambda^\star = \arg\max_{\lambda} P(O|\lambda)$$

This is by far the most difficult of the three HMM problems because there is no known analytical method that maximizes the model parameters in this way. Instead, the problem can be solved using the iterative *Baum-Welch* algorithm, also known as the *forward-backward* algorithm.

First, in a manner similar to the forward probability, we define backward probability as:

$$\beta_t(i) = \begin{cases} 1 & \text{if } t = T \\ \sum_{i=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(i) & \text{if } t < T \end{cases}$$

where $\beta_t(i)$ is the probability of generating partial observation O_{t+1}^T (from $t+1$ to the end) given that the HMM in state i at time t.

With help of the terms α and β, it is possible to calculate the probability $\gamma_t(ij)$ of taking the transition from state i to state j at time t, given the model and observation sequence:

$$\gamma_t(j) = P(q_t = S_j | O, \lambda) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

To reestmate a_{ij}, we have to find the ratio between expected number of transitions from state i to state j and the expected total number of transitions

from state i. It turns out that the reestimation $\hat{a}_{ij}$ can be calculated from $\gamma_t(ij)$ as:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} \gamma_t(i,j)}{\sum_{t=1}^{T} \sum_{k=1}^{N} \gamma_t(i,k)}$$

To recompute the observation probabilities, we need to find the ratio between expected number of times the observation data emitted the symbol v_k when being in state j, and the expected total number of times in state j. This reestimation can be also be calculated with help of $\gamma_t(ij)$:

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

The entire training procedure of HMMs first chooses some estimate for model parameters a and b, and then uses the given equations to reestimate the parameters. The Baum-Welch algorithm guarantees a monotonic likelihood improvement on each iteration, and eventually the likelihood converges to a local minimum.

2.3.2 Selection of basic units

The HMMs can be used to provide the estimates of $P(O|W)$ in speech recognizers. For isolated word recognition with sufficient training data it may be possible to build a HMM for each word. However, for continuous large vocabulary recognition it is unlikely that there is sufficient data for training each word in the lexicon. Instead, some sub-word units have to be used. The selected sub-word units should be accurate, trainable and generalizable: the units should be able to represent the acoustic realizations in different contexts; there should be enough training data to robustly estimate the parameters of the unit; and any new word should be derivable from the predefined inventory of trained units. The majority of modern speech recognition systems use sub-word units called *phones*. Phones usually correspond to one or more phonemes in the underlying language and usually also include silence, short pause and some noise models. The chosen phone set depends on the availability of sufficient training data and other practical issues. The HMMs corresponding to the phones may then be concatenated to form composite word models and sentence models, as shown in Figure 2.6.

2.3.3 Clustered context-dependent acoustic units

When one HMM is trained for each basic phone, it is referred to as a monophone or context independent system. However, there is a large amount of variation between realizations of the same phone depending on its neighborhood. This effect is called co-articulation and it happens due to the inertia restricting any abrupt movement of the articulatory organs. A widely used approach to improve accuracy and trainability is using context dependent units. Phones that take into

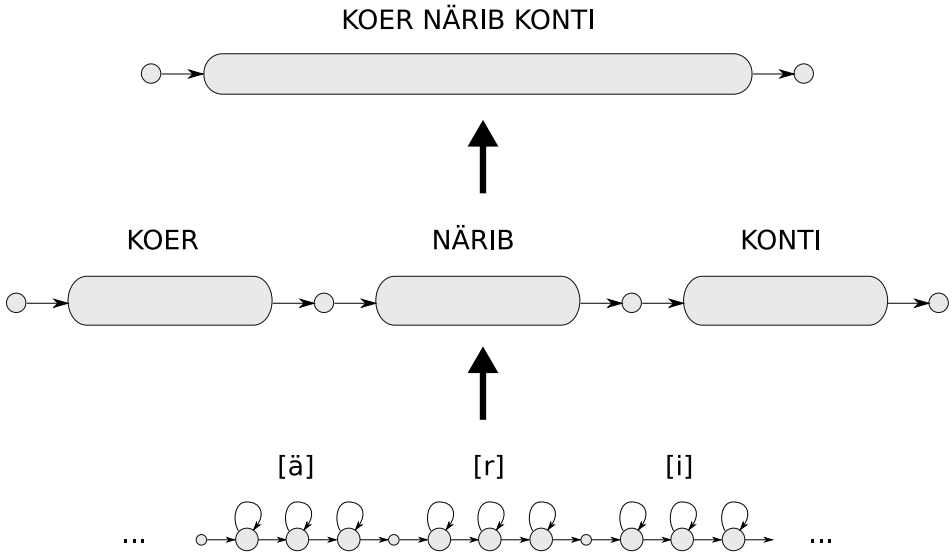

Figure 2.6: Construction of a composite sentence HMM from word HMMs and phone HMMs, for a sentence *"koer närib konti"*, "dog eats a bone".

account the immediate left and right neighboring phones are called *triphones*. If two phones have the same identify but different left and/or right contexts, they are considered different triphones.

The number of states and model parameters of an acoustic model consisting of triphones is significantly larger than those of a monophone system. It is therefore unlikely that there is enough training data to reliably estimate all different triphones in the training data. Furthermore, it is very common that many triphones that are needed for composing word models of the final system do not occur in the training data at all. The most common solution to this problem is to share some of the model parameters by sharing the state conditional observation densities among different models. The rationale behind this approach is that many phones that are produced in articulatory similar way have a similar effect on the neighboring phones. For example, /m/ and /n/ are both nasals and have a similar effect on the neighboring vowels, thus it might be a good idea to share the first states of triphones that correspond to same vowel but represent the left /m/ and /n/ context. This leads to a much more manageable number of models that can be trained, and in addition, triphones that are not seen in the training data but are needed for recognition can be synthesized from existing states. Figure 2.7 shows a hypothetical clustering of some triphones.

The states that are to be shared are often determined in a data-driven manner using phonetic decision trees [Young et al., 1994]. The phonetic decision trees classifies triphone states of triphones that are represented in the training data by asking binary linguistic questions about the context of the triphone. The

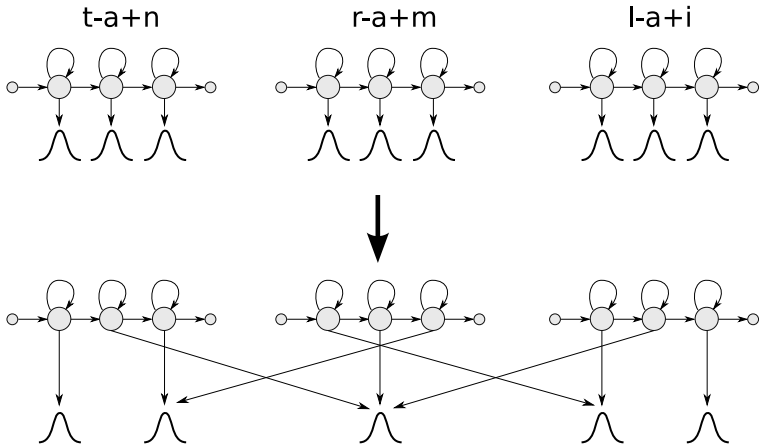

Figure 2.7: Hypothetical clustering of nine triphone state observation densities into five shared densities.

questions can be simple categorical questions (e.g. "*Is the left context a nasal?*" or conjunctions, disjunctions and/or negations of the simple questions. The set of the simple categorical questions is designed by an expert. The decision tree is composed automatically by splitting the data using the question that provides the largest likelihood increase for the acoustic models against training data. The splitting will terminate in the final leaves or if the number of training data samples per state falls below a set threshold.

The set of linguistic categories that are used for forming the questions can be also generated automatically so that maximally separated partitioning are ensured [Singh et al., 1999]. This method has an important advantage of extensibility to languages for which the phonetic structure is not well understood by the system designer.

2.4 Language Modelling

While the acoustic model is used to compute the likelihood of a certain word sequence given the measured acoustic evidence, the language model helps to estimate *a priori* probability of the word sequence, i.e.:

$$\hat{P}(W), \;\; W = w_1 w_2 ... w_n$$

where the caret denotes the estimate of the probability. This probability helps the speech recognizer in deciding upon one of possibly several acoustically similar, competing ways of segmenting the sequence of observation vectors into words according to their prior likelihood. It also helps to dramatically constrain the search space of possible word sequences.

2.4.1 N-gram language models

It is clear that it's impractical to generate and store probabilities for all possible word sequences in the language. The most widely used statistical language model, the so-called *n*-gram model, uses the fact that the probability of a word sequence can be decomposed into a product of conditional probabilities, using the chain rule

$$P(w_1, ...w_n) = \prod_{i=1}^{n} P(w_i|w_1, w_2, ..., w_{i-1})$$

where $P(w_i|w_1, w_2, ..., w_{i-1})$ is the probability that w_i will follow, given that the word sequence $w_1, w_2, ..., w_{i-1}$ (also referred to as word history) appeared previously. The sequential manner of this decomposition makes it particularly appropriate to the way how search for the most probable word sequence is carried out in a speech recognizer. However, for a vocabulary size of v there are v^{i-1} different possible histories and to specify $P(w_i|w_1, w_2, ..., w_{i-1})$ completely, v^i probabilities would have to be estimated. In reality, such probabilities are impossible to estimate for even moderate values of i since they would need huge amounts of storage, and more importantly, such probabilities cannot be simply estimated from training data since most histories $w_1, w_2, ..., w_{i-1}$ occur never or only very rarely. As a solution, one can make a (possibly false) assumption that $P(w_i|w_1, w_2, ..., w_{i-1})$ depends only on some equivalence classes. The most obvious word history equivalence classification is a simple truncation of the word history to the last N words, which leads to the *n*-gram language model. The equivalence class definition of the *n*-gram language model is that all word histories which end in the same $N - 1$ words are identical from the language modelling point of view, i.e.:

$$P(w_1, ...w_n) = \prod_{i=1}^{n} P(w_i|w_1, w_2, ..., w_{i-1}) \approx \prod_{i=1}^{n} P(w_i|w_{i-2}, w_{i-1})$$

Truncating the word history in such manner reduces the number of parameters so that they can be more robustly estimated while still preserving the usefulness for estimating the likelihood of the current word.

The most obvious drawback of *n*-gram language models is that they don't model long-distance relationships. E.g. given the Estonian sentence *Koer sööb murul lamades konti* "Dog eats a bone while lying on the mown" and a trigram language model, the probability of the word *konti* is conditioned only on the two previous words *murul lamades* "while lying on the mown" while in reality the strong relationship between the words *koer sööb* "dog eats" and *konti* is very clear.

Another weakness of *n*-gram models is that they do not disambiguate sentences which are grammatically incorrect. For example, given a reference utterance

```
kongi põrand oli nii tihedasti asustatud et konngi
ei mahtunud enam kongi
```

and the recognized word sequence

```
kongi põrandale nii tihedasti asustatud et kongi
mahtunud enam kongi
```

it can be seen that each triplet of the recognized sentence is entirely plausible but the resulting word sequence does not make much sense [1].

Despite the drawbacks, bigram and especially trigram language models are still the most effective and widely used language models in contemporary speech recognition systems. Their effectiveness lies in the way model parameters can be estimated from training corpora and the simplicity with which they can be incorporated into a speech recognition search process. The local relationships that *n*-gram models capture seem to be most important for most languages.

N-gram estimation

The trigram probability $P(w_i|w_{i-2}, w_{i-1})$ can be estimated by simply counting the occurrences of the triplet $w_{i-2}w_{i-1}w_i$ and normalize it:

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}w_{i-1}w_i)}{\sum_{w_i} C(w_{i-2}w_{i-1}w_i)} = \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})}$$

The text available for building a model is called training corpus. For N-gram models, the size of the corpus is typically tens or hundreds of millions of words. The estimate for $P(w_i|w_{i-2}, w_{i-1})$ given above is called the maximum likelihood (ML) estimate of $P(w_i|w_{i-2}, w_{i-1})$ since this assignment of probabilities yields a trigram model that assigns the highest probability to the training corpus among all possible trigram models.

The maximum likelihood estimate assigns a zero probability to all N-grams that never occur in the training corpus. However, it is very common that many perfectly valid trigrams never occur in the training data. For example, given an Estonian training corpus of over 60 million words, over 30% of all trigrams in the handout texts never occur in the training corpus, even when using morphemes as basic units for language modelling. If a certain trigram is assigned a zero probability by a language model, the whole sentence that contains that trigram is never considered as a candidate for possible transcription regardless of how unambiguous the acoustic signal is. Assigning all trigrams a non-zero probability helps prevent errors like this in speech recognition. Modification of maximum likelihood probabilities to allow estimation of probabilities for unseen events is

[1] In reality, the subword units are used as basic units in Estonian speech recognizer and the actual recognized sequence is: `kongi põranda _le nii tihedasti asusta _tud et kongi mahtu _nud enam kongi`

generally referred to as smoothing. In the following section, review of Good-Turing estimates, Katz's backoff smoothing and Kneser-Ney smoothing is given.

Good-Turing estimates

The Good-Turing estimate [Good, 1953] is a smoothing technique to deal with infrequent N-grams. It is a central method for many smoothing techniques.

The Good-Turing estimate states that for any N-gram that occurs r times, we should modify the count by a discount coefficient d_r where

$$d_r = (r+1)\frac{n_r+1}{rn_r}$$

and where n_r is the number of N-grams that occur exactly r times in the training corpus. The probability of an N-gram α occurring r times can be then estimated by normalizing the pseudo-count $d_r r$:

$$P_{GT}(\alpha) = \frac{d_r r}{N}$$

where

$$N = \sum_{r=0}^{\infty} n_r d_r r = \sum_{r=0}^{\infty} (r+1)n_{r+1} = \sum_{r=1}^{\infty} rn_r$$

i.e., N is equal to the original number of counts in the distribution.

Katz [Katz, 1987] suggests that events that occur more than k times (where k is typically in the range of 5 to 8) can be reliably estimated by their relative frequencies, so they are not discounted. Hence the discount coefficient is defined as

$$d_r = \begin{cases} \frac{(r+1)\frac{n_r+1}{rn_r} - (k+1)\frac{n_{k+1}}{n_1}}{1-(k+1)\frac{n_{k+1}}{n_1}} & \text{if } 1 \leq r \leq k \\ 1 & \text{if } r < k \end{cases}$$

The relative values of the count-of-counts must satisfy the relationships

$$n_1 \geq 2n_2 \geq 3n_3 ...$$

otherwise the resulting discounted counts will not be consistent with each other. For most naturally occurring data, these constraints are satisfied.

Katz backing-off scheme

Katz back-off smoothing technique [Katz, 1987] extends the Good-Turing discounting scheme by adding the combination of higher-order models with lower order models. In general, backing off refers to using a probability estimate proportional to one from a more general distribution when the estimate from the

specific distribution is missing or unreliable. In the case of an N-gram language model, the back-off technique applies a more general $(N-1)$-gram distribution if the N-gram is deemed unreliable. For example, a trigram model provides a greater refinement in predictive power over bigram and unigram models. If a trigram probability for a certain N-gram is unreliable or cannot be estimated, the bigram model is used instead. In turn, when the bigram is unreliable, the unigram model is applied. Usually, an N-gram estimate is deemed unreliable if it is missing in the language model, which is a result from the N-gram being missing in the training corpus or it occurring less than a certain number (cutoff) of times.

The Katz back-off technique is illustrated using a trigram model as an example. Given an estimate for the probability of observed events $\hat{P}$, the total probability of unseen events $\hat{P}_u$ occurring in the context $w_{i-2}w_{i-1}$ is given by

$$\hat{P}_u(w_{i-2}, w_{i-1}) = 1 - \sum_{w:N(w_{i-2},w_{i-1},w_i)>0} \hat{P}(w|w_{i-2}, w_{i-1}).$$

The backing-off scheme is used to distribute the probability of unseen trigrams among all unobserved trigrams according to the more general bigram distribution $\hat{P}(w_i|w_{i-1})$:

$$\hat{P}(w|w_{i-2}, w_{i-1}) = \frac{\hat{P}_u(w_{i-2}, w_{i-1})}{\sum_{w:N(w_{i-2},w_{i-1},w_i)=0} \hat{P}(w|w_{i-1})} \hat{P}(w|w_{i-1})$$

The denominator ensures that the probabilities sum to one.

By combining the backing-off technique with discounting scheme that uses discount coefficient d_r for all counts of r, the trigram estimate is given as

$$\hat{P}(w_i|w_{i-2}, w_{i-1}) = \begin{cases} d_{C(w_{i-2},w_{i-1},w_i)} \frac{C(w_{i-2},w_{i-1},w_i)}{C(w_{i-2},w_{i-1}} & \text{if } C(w_{i-2}, w_{i-1}, w_i) \geq 1 \\ \alpha(w_{i-2}, w_{i-1}) \hat{P}(w_i|w_{i-1}) & \text{if } C(w_{i-2}, w_{i-1}, w_i) = 0 \end{cases}$$

where $\alpha(w_{i-2}, w_{i-1})$ is a back-off weight which is defined as

$$\alpha(w_{i-2}, w_{i-1}) = \frac{1 - \sum_{w:C(w_{i-2},w_{i-1},w)>0} \hat{P}(w|w_{i-2}, w_{i-1})}{1 - \sum_{w:C(w_{i-2},w_{i-1},w)>0} \hat{P}(w|w_{i-1})}$$

and ensures that the probabilities $\hat{P}(w|w_{i-2}, w_{i-1})$ sum to one. In the last expression, the numerator corresponds to the left-over probability mass obtained from discounting the counts of observed events and the denominator is a normalizing factor that expresses the total back-off probability. Smoothing is applied recursively with the unigram probabilities being smoothed first since these are required for bigram smoothing, etc.

The Katz backing-off method is computationally efficient since all the back-off weights can be estimated during language model training. Also, the scheme can be relatively easily incorporated into the decoding process.

Absolute discounting

Absolute discounting [Ney et al., 1994] is an improvement over simple interpolation of higher-order and lower-order models. It involves subtracting a fixed discount $D \leq 1$ from each non-zero count:

$$\begin{aligned} & P_{abs}(w_i|w_{i-n+1}...w_{i-1}) \\ & = \frac{\max(C(w_{i-n+1}...w_1) - D, 0)}{\sum_{w_i} C(w_{i-n+1}...w_1)} \\ & + (1 - \lambda_{w_{i-n+1}...w_{i-1}}) P_{abs}(w_i|w_{i-n+2}...w_{i-1}) \end{aligned}$$

To make this distribution sum to one, the following condition must be satisfied:

$$1 - \lambda_{w_{i-n+1}...w_{i-1}} = \frac{D}{\sum_{w_i} C(w_{i-n+1}...w_1)} \mathbb{C}_{1+}(w_{i-n+1}...\bullet)$$

where

$$\mathbb{C}_{1+}(w_{i-n+1}...\bullet) = |\{w_i : C(w_i|w_{i-n+1}...w_i) > 0\}|$$

The notation $\mathbb{C}_{1+}$ is meant to evoke the number of unique words that follow the history $w_{i-n+1}...w_{1-1}$. The discount D is suggested to be taken as

$$D = \frac{n_1}{n_1 + 2n_2}$$

where n_1 and n_2 denote the total number of N-grams with exactly one and two counts, respectively.

Kneser-Ney smoothing

Kneser and Ney [Kneser and Ney, 1995] propose an improvement to absolute discounting technique that combines lower-order distribution with a higher-order distribution in a novel manner. In previous algorithms, the lower-order distribution that is used in the backing-off scheme is usually a smoothed version of the lower-order maximum likelihood distribution. However, the lower-order distribution is significant only when few or no counts are present in the higher-order distribution. The idea of Kneser-Ney smoothing is to optimize the lower-order distribution to perform well in these situations, i.e., when the Katz backing-off scheme actually refers to it.

The need for such approach becomes clearer when we consider a bigram model on a data where there exists a word that is very common, e.g. *Francisco*,

but which only or mostly occurs after a single word, e.g. *San*. Since the frequency of *Francisco* is high, the unigram probability $P(\text{Francisco})$ will be high, and an algorithm such as Good-Turing discounting will assign a relatively high probability to the word *Francisco* occurring after previously unseen bigram histories. However, perhaps the unigram probability of this word should not be high since there is reliable evidence that it mostly occurs after the word *San* in which case the bigram distribution already models its probability well.

According to this line of reasoning, the unigram distribution should be proportional not to the number of occurrences of a word, but to the number of words that it follows. When traversing the training data and building a bigram model to predict the current words based on the already traversed data, the unigram probability of the word is the decisive factor only if the current bigram hasn't occurred already in the training data. If we assign a count to the current word whenever we actually have to refer to the unigram probability, the number of counts assigned to each word will be simply the number of different contexts that it follows.

Kneser-Ney smoothing uses the same general formula as absolute discounting:

$$\begin{aligned}
&P_{KN}(w_i|w_{i-n+1},...,w_{i-1}) \\
&= \frac{\max(C(w_{i-n+1}...w_1) - D, 0)}{\sum_{w_i} C(w_{i-n+1}...w_1)} \\
&+ \frac{D}{\sum_{w_i} C(w_{i-n+1}...w_1)} \mathbb{C}_{1+}(w_i|w_{i-n+1}...\bullet) P_{KN}(w_i|w_{i-n+2}...w_{i-1})
\end{aligned}$$

However, the lower-order probabilities are computed differently:

$$P_{KN}(w_i|w_{i-n+2},...,w_{i-1}) = \frac{\mathbb{C}_{1+}(\bullet w_{i-n+2}...w_i)}{\mathbb{C}_{1+}(\bullet w_{i-n+2}...w_{i-1}\bullet)}$$

where

$$\mathbb{C}_{1+}(\bullet w_{i-n+2}...w_i) = |\{w_{i-n+1} : C(w_{i-n+1}...w_i) > 0\}|$$

and

$$\mathbb{C}_{1+}(\bullet w_{i-n+2}...w_{i-1}\bullet) = |\{(w_{i-n+1}, w_i) : C(w_{i-n+1}...w_i) > 0\}|$$

i.e., $\mathbb{C}_{1+}(\bullet w_{i-n+2}...w_i)$ is the number of different words that precede $w_{i-n+2}...w_i$ and $\mathbb{C}_{1+}(\bullet w_{i-n+2}...w_{i-1}\bullet)$ is the number of different word pairs that surround the context $w_{i-n+2}...w_{i-1}$.

It can be shown that the formula for lower-order estimates can be derived if we select the lower-order distribution such that the marginals for the higher-order smoothed distribution match the marginals of the training data, i.e. for a bigram

model the following constraint should be satisfied:

$$\sum_{w_{i-1}} P_{KN}(w_{i-1}w_i) = \frac{C(w_i)}{\sum_{w_i} C(w_i)}$$

A modification of Kneser-Ney smoothing [Chen and Goodman, 1998] is widely used and it usually exhibits excellent performance. Instead of using single discount D for all nonzero counts as in the original method, we can have three different parameters, D_1, D_2, and D_{3+} that are applied to N-grams with one, two and three or more counts, respectively:

$$\begin{aligned}
&P_{KN}(w_i|w_{i-n+1},...,w_{i-1}) \\
&= \frac{\max(C(w_{i-n+1}...w_1) - D(C(w_{i-n+1}...w_1)), 0)}{\sum_{w_i} C(w_{i-n+1}...w_1)} \\
&+ \gamma(w_{i-n+1}...w_1) P_{KN}(w_i|w_{i-n+2}...w_{i-1})
\end{aligned}$$

where

$$D(c) = \begin{cases} 0 & \text{if } c = 0 \\ D_1 & \text{if } c = 1 \\ D_2 & \text{if } c = 2 \\ D_{3+} & \text{if } c \geq 3 \end{cases}$$

The optimal values for D_1, D_2, and D_{3+} have been estimated to be

$$\begin{aligned}
D_1 &= 1 + 2Y\tfrac{n_2}{n_1} \\
D_2 &= 2 - 3Y\tfrac{n_3}{n_2} \\
D_{3+} &= 3 - 4Y\tfrac{n_4}{n_3}
\end{aligned}$$

where

$$Y = \frac{n_1}{n_1 + 2n_2}$$

2.4.2 Language model evaluation

The best way to evaluate performance of a language model is to incorporate it into a speech recognition system, perform a recognition experiment and look at the word error rate. The true quality of the language model can only be measured in this way, since its utility is implicitly linked to the behavior of the acoustic model. However, although the performance of the language model inside the recognition system is ultimately crucial, it is impractical to evaluate language models in this way due to the high computational cost of recognition experiments. Therefore, there is a need for evaluating language models in isolation from the acoustic model, as quickly and objectively as possible, and in a way that corresponds to the performance of the language model when it would be used in a speech recognizer.

An information theoretic measure that satisfies those conditions will be introduced next.

Entropy and perplexity

The task of the language model in the speech recognition system is to compute *a priori* probabilities of different word sequences. The probability that is assigned to a test text that did not form part of data that was used to train the model, gives some indication of how well the model can predict sentences of the given language. The higher the probability of the test text, the better the model is in predicting this text.

A natural language can be regarded to have been produced by an information source that emits symbols $z(i)$ at discrete time intervals $i \in 0, 1, ..., \infty$ from a certain finite set according to some statistical law. The symbols that are produced might be, for example, words, word sequences, or some sub-word units. Let the emission of a symbol be referred to as event. Assuming that the source emits a sequence $z(1), ..., z(n)$ with a probability $P(z(1), ..., z(n))$, the per-event self-information of the sequence is [Shannon, 1953]

$$I_s(z(1), ..., z(n)) = -\frac{1}{n} \log(P(z(1), ..., z(n)))$$

The self-information is an information theoretic measure of the amount of information gained by observing the sequence $z(1), ..., z(n)$ and thereby removing the uncertainty about it. Rare sequences, i.e. those with low probability, carry a larger amount of information than more frequent sequences. The per-event entropy, or the average per-event self-information of the source is then:

$$h = -\lim_{n\to\infty} \frac{1}{n} \sum_{z(1),...,z(n)} P(z(1), ..., z(n)) \log P(z(1), ..., z(n)))$$

where the summation is over all possible event sequences of length n that source is capable of producing. Entropy is the average measure of the amount of information contained in the set of sequences the source may produce. If the source is ergodic[2] then the entropy is equivalent to

$$h = -\lim_{n\to\infty} \frac{1}{n} \log P(z(1), ..., z(n)))$$

The per-event entropy of the source is a measure of the uncertainty that the source experiences in determining an event. However, since the true mechanism of the natural language source is unknown, only the approximation of the probabilities $P(z(1), ..., z(n))$ can be calculated. Furthermore, the length of the sequences

[2] A source is said to be *ergodic* if its statistical characteristics can be determined over a sufficiently long sequence *temporally*, instead of from the *ensemble* of sequences.

$z(1), ..., z(n)$ at our disposal is always finite in practical situations, so that we must approximate the above equation by:

$$\hat{h} = -\frac{1}{n} \log \hat{P}(z(1), ..., z(n))$$

Thus, the entropy of the source is approximated by the average per-event log probability of the observation. In the language modelling framework, this observation is the training corpus. It can be shown that the approximated entropy is always greater than or equal to the actual entropy since the probability estimates of the event sequences can never be better than the actual probabilities. This is very appealing because it implies that only a perfect model can assign the highest probability to an actually observed sequence and any imperfections in the model will lead to lower probability. Therefore, the probability that the model assigns to an actually observed sequence can be taken as a measure of the model quality. To make this measure independent of the length of the sequence, a per-event average may be used, which is called perplexity:

$$PP = 2^{\hat{h}} = (\hat{P}(z(1), ..., z(n)))^{-\frac{1}{n}}$$

The perplexity may be interpreted as the average branching factor at every time instant according to the source model. In language modelling terms, this corresponds the average number of equally probable words that follow any given word. For example, if the language model has a vocabulary of size K and every word is equally probable in any context, the probability estimate of a word sequence of length N is

$$\hat{P} = \prod_{N} \frac{1}{K} = \frac{1}{K^N}$$

and the perplexity is

$$PP = \frac{1}{K^N}^{-\frac{1}{N}} = K$$

Of course, the perplexity cannot take into account the acoustic difficulty in distinguishing a word. It is possible that a language model that can differentiate well between acoustically similar words may result in a better word error rate than a model with a lower perplexity, when incorporated into a speech recognition system. Different improved quality measures have been proposed [Jelinek et al., 1992, Chelba, 2006] that take into account this aspect but those measures have the disadvantage of becoming specific to the nature of the acoustic model being used.

It should be also noted that it is only sensible to compare language models based on the same test text, and using the same language model vocabulary. For example, it is easy to construct a language model for Estonian that has a perplexity

as low as 33, if we set the vocabulary to include all letters in the Estonian alphabet and a space delimiter, and assign all units a uniform probability. Such language model can model any Estonian text but of course doesn't work well in a speech recognition system, since many letters are acoustically highly confusable.

Chapter 3

Properties of the Estonian language

The Estonian language belongs to the Balto-Finnic subgroup of Finno-Ugric languages. It has about 1.1 million native speakers. Among other larger Finno-Ugric languages, Estonian is closely related to Finnish, and more distantly to the Hungarian language of the Ugric branch. Over the course of Estonian history, German has had a strong influence on Estonian, both in vocabulary and syntax.

In this chapter, an overview of Estonian phonology, orthography, morphology and syntax is given.

3.1 Phonology

This section describes the distinctive sounds within the Estonian language. The vowel and consonant inventory is listed. An approach to handling long phonemes and diphthongs is described.

The last subsection gives a modern treatment of the three-way quantity opposition in Estonian words.

3.1.1 Vowels

Estonian has nine vowels, each of which corresponds to a single grapheme. They can be grouped with respect to the tongue position, tongue height and the roundedness of the lips in their articulation process. All vowels are shown in table 3.1, together with their symbols according to IPA and SAMPA representation. There is no single appropriate character for the Estonian /õ/ in the IPA inventory, although [ɤ] seems to be the most common [Eek and Meister, 1999].

All short and long (double) vowels occur in the primary-stressed first syllable. Beyond the primary-stressed syllable the distribution of vowels is more restricted: in non-initial syllables only short /a/, /e/, /i/, /u/, /o/ occur, whereas /o/ is used in

Table 3.1: Estonian vowel inventory [Eek and Meister, 1999].

IPA	*SAMPA*	*Estonian phono-logical transcrip-tion*	*Example word*	*Tongue position, tongue height, lips*
ɑ	A	/a/	*sada*	back, low, unrounded
e	e	/e/	*keda*	front, medium-high, un-rounded
i	i	/i/	*kilu*	front, high, unrounded
o	o	/o/	*pori*	back, medium-high, rounded
u	u	/u/	*kuri*	back, high, rounded
ɤ, ɯ, ə	7	/õ/	*kõma*	back, high, unrounded
æ	{	/ä/	*käru*	front, low, unrounded
ø	2	/ö/	*löma*	front, medium-high, rounded
y	y	/ü/	*mürin*	front, high, rounded

late loanwords, names and in foreign words. Long vowels do not occur in non-initial syllables of native Estonian words. However, long vowels do occur in the primary or secondary-stressed non-initial syllables of foreign words.

There is little difference in the quality of short and long vowels, therefore it is not justified to define them as different phonemes [Eek and Meister, 1999]. A long duration is designated by sequences of two identical segmental phonemes.

There are 36 segmental diphthongs and polysyllabic vowel clusters are also found. All nine vowels are used as the first component of a diphthong and only /a/, /e/, /i/, /u/ and /o/ as the second component, as seen in table 3.2.

In the primary-stressed first syllable of native words and older loanwords, only 25 diphthongs occur. From those, 17 appear in the long vs. ”overlong” opposition; 8 dipthongs occur only in the overlong foot.

In non-initial syllables of native words, only the dipthongs *ai*, *ei* and *ui* occur.

Usually, diphthongs are formed so that the main quality of the vowels is retained [Kraut, 2000]. In case of diphthongs *ea*, *oa* and *öa*, there is a tendency to pronounce the first component slightly higher than the corresponding vowel.

Some native words contain vowel clusters consisting of three phonemes (e.g. *kaua*, *viie*, *laiem*). In such cases, a syllable border is placed before the last vowel in the cluster and the second syllable is linked by pronouncing /w/ (after a long *uu* or *u*-final diphthong) or /j/ (after long *ii* or *i*-final diphthong) between the two syllables: *kaua* [kɑuwɑ], *viie* [viije], *laiem* [lɑijem]. However, in such cases, [w] and [j] do not have phonemic status since their occurrence is always predictable

Table 3.2: Estonian diphthongs [Eek and Meister, 1999]. Diphthongs marked with * appear only in the overlong foot; diphthongs marked by ** occur only in foreign words.

	i	e	u	o	a
i		ie**	iu	io**	ia**
e	ei		eu**	eo*	ea
ä	äi	äe	äu	äo	
ü	üi**	üe**		üo**	üa**
ö	öi	öe*			öa*
u	ui	ue**		uo**	ua**
o	oi	oe	ou		oa*
õ	õi	õe	õu	õo	õa*
a	ai	ae	au	ao**	

and is considered to be a coarticulation effect.

3.1.2 Consonants

The consonant phonemes together with their articulatory characteristics are listed in table 3.3. One of the main features of the Estonian consonant system is that there is no voiced-unvoiced opposition (/b/ – /p/, /z/ – /s/). This is replaced by pronouncing the same voiceless phoneme in different lengths (/p/ – /pp/, /s/ – /ss/). The short plosives b̥, d̥, g̥ are considered half-voiced, or according to more recent conventions [Eek and Meister, 1999, Kraut, 2000], they are just short versions of the corresponding unvoiced plosives. Voicing of the beginning of short plosives between the first and second syllable covers a segment which is about 3/10 of the plosive's whole duration. In the non-initial syllables of longer words the partial voicing of a short plosive is even more extensive; in spontaneous speech short plosives are often fully voiced in voiced context. The spreading of foreign language knowledge has increased the number of speakers who pronounce the short plosive completely voiced.

There are some secondary (non-phonemic) segmental units that are considered positional variants of the main phonemes (table 3.4). The phoneme /n/ is realized as a palato-velar nasal [ŋ] before palato-velar plosives (except when there is a morpheme boundary between /n/ and /k/). Sonorants preceded by /h/ are devoiced at the end of one-syllable words. Sometimes the sonorant remains voiced but then they are syllabic consonants [Eek and Meister, 1999].

In word-initial position, the short/long opposition of plosives has been neutralized: the word-initial plosive is relatively long in absolute word beginning, or short if occurring in intra-sentence context. The word-initial *g*, *b*, *d* occur only in foreign words and are pronounced just like *k*, *p*, *t*: *baar* /paa:r/ 'bar', nom. sg. – paar /paa:r/ 'pair', nom.sg.; *gaas* /kaa:s'/ 'gas', nom.sg. – *kaas* /kaa:s/ 'cover',

Table 3.3: Estonian consonant inventory [Eek and Meister, 1999].

IPA	*SAMPA*	*Estonian phonological transcription*	*Example word*	*Type of articulation, voiced/voiceless, place of articulation*
b̥, p	p	/p/	*taba*	plosive, voiceless, bilabial
d̥, t	t’	/t/	*padu*	plosive, voiceless, denti-alveolar
d̥ʲ, tʲ	t, t’	/t’/	*padi*	palatalized plosive, voiceless, denti-alveolar
g̥, k	k	/k/	*kagu*	plosive, voiceless, palato-velar
f	f	/f/	*foori*	fricative, voiceless, labiodental
v	v	/v/	*kava*	fricative, voiced, labiodental
s	s	/s/	*mäsu*	fricative, voiceless, alveolar
sʲ	s’	/s’/	*kasi*	palatalized fricative, voiceless, alveolar
ʃ	S	/š/	*šefi, loožі*	fricative, voiceless, postalveolar (usually labialized)
h	h	/h/	*sahin*	fricative, voiceless, glottal-oral; short degree - voiced lenis; long degree - voiceless geminate
m	m	/m/	*samu*	nasal, voiced, bilabial
n	n	/n/	*kanu*	nasal, voiced, alveolar
nʲ	n’	/n’/	*pani*	palatalized nasal, voiced, alveolar
l	l	/l/	*kalas*	lateral, voiced, alveolar-postalveolar
lʲ	l’	/l’/	*pali*	palatalized lateral, voiced, alveolar-postalveolar
r	r	/r/	*nari*	thrill, voiced, alveolar
j	j	/j/	*maja*	approximant, palatal

Table 3.4: Secondary (non-phonemic) segmental units [Eek and Meister, 1999].

IPA	*SAMPA*	Example	Description
ŋ	N	pank [pɑŋk]	/n/ is realized as ŋ before /k/ if the velar plosive is not preceded by morpheme boundary
m̥	m_0	vihm [vihːm̥]	at the end of one-syllable words sonorants preceded by /h/ (and other obstruents) are idiosyncratically devoiced
n̥	n_0	pahn	
l̥	l_0	kahl	
r̥	r_0	kõhr	
ṿ	v_0	kahv	
m̩	=m	vihm [vihːm̩]	if at the end of one-syllable words sonorants preceded by /h/ (or other obstruents) remain voiced, then these sonorants are syllabic consonants
n̩	=n	pahn	
l̩	=l	kahl	
r̩	=r	kõhr	
v̩	=v	kahv	
w	w	kaua [kɑuwɑ]	between long /u/ or /u/-final diphthong and the following short /a/ and /e/ some speakers pronounce [w]

nom.sg.; duur /tuu:r/ 'major key', nom.sg. – *tuur* /tuu:r/ 'sturgeon; ice pick', nom.sg., etc. However, there is a tendency of pronouncing word-initial *g*, *b*, *d* as voiced.

The phonemes /f/ and /š/ occur in foreign words; they are not fully adapted to the phonological system of Estonian. The orthographically exposed *z* and *ž* in foreign words are usually pronounced as /s/ and /š/, respectively.

There are no affricate phonemes in Estonian because in the clusters ts and tš behave as geminates and other consonant clusters: the syllable boundary between consonants is clearly determinable, e.g. putši /put'ši/ 'minor revolt', gen.sg. – /put':ši/ 'minor revolt', part.sg.

Primary place of articulation of palatalized consonants /t'/, /s'/, /n'/, /l'/ is not much different from their non-palatalized equivalents; only the back boundary of the front contact area on palatograms is shifted somewhat backwards. The main difference lies in secondary articulation, i.e. in the side contact areas due to /i/-like final transition of the preceding vowel [Eek and Meister, 1999].

3.1.3 Quantity degrees

One special feature of Estonian phonology that has caused a great amount of interest among linguists and phoneticians is the three-way quantity opposition. Traditional treatment is based on three distinctive degrees of segmental duration. All vowels and consonants can occur in short, long and overlong duration degrees

[Ariste, 1939, Ariste, 1953].

According to the modern treatment quantity degrees are foot-level phenomena and can't be explained on segmental or syllabic levels. Quantity degrees are described as duration ratios between the first and second syllable (see table 3.5), which have been found to be very stable even in spontaneous speech [Engstrand and Krull, 1994].

Table 3.5: Duration ratio of the first and second syllable, depending on the quantity degree of the foot.

Source	Q1	Q2	Q3
[Lehiste, 1960]	0.7	1.5	2.0
[Liiv, 1961]	0.7	1.6	2.6
[Eek, 1974]	0.7	2.0	3.9
[Krull, 1991]	0.5-0.7	1.2-2.1	2.2-2.9

Furthermore, quantity degree is a complicated foot pattern the identification of which depends on the total effect of several simultaneous features: duration ratio, position of the fundamental frequency (F0) peak, and distribution of acoustic energy in bisyllabic sequencies [Eek and Meister, 1999]. A foot is in the first quantity (Q1) (e.g. *kalu* /koli/ 'fish', part. pl.) when the stressed syllable of the foot ends in a short vowel, and the short vowel of an unstressed second syllable is phonetically halflong or long. The fundamental frequency is rising, F0 peak is at the end of a voiced rhyme of the stressed syllable; in an unstressed syllable F0 is falling. A foot is in the second quantity (Q2) (e.g. *(selle) kaalu* /kaalu/ 'weight', gen. sg.) when its stressed syllable is long (i.e. when it ends in a long vowel, diphthong or at least one consonant) and the vowel of an unstressed second syllable is phonetically short without qualitative reduction. The fundamental frequency peak is in the second half of a voiced rhyme of the stressed syllable (rising or level tone); in an unstressed syllable F0 is falling. A foot is in the third quantity (Q3) (e.g. *(seda) kaalu* /kaa:lu/ 'weight', part. sg.) when its stressed syllable is long and the vowel of an unstressed syllable is extra short, weakened and qualitatively reduced [Eek and Meister, 1997]. The F0 peak is in the first half of a voiced rhyme of the stressed syllable following by falling tone which continues in an unstressed syllable.

3.2 Orthography

Estonian uses the Latin alphabet and consists of 32 letters: *a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*, *i*, *j*, *k*, *l*, *m*, *n*, *o*, *p*, *q*, *r*, *s*, *š*, *z*, *ž*, *t*, *u*, *v*, *w*, *õ*, *ä*, *ö*, *ü*, *x*, *y*. The letters *f*, *š*, *z*, *ž* and *f* are used only in foreign and loan words and foreign names. The letters *c*, *q*, *w*, *x* and *y* are used only in foreign names.

Estonian orthography is largely phonetic with each phoneme of the language

represented by exactly one grapheme. However, there are many important exceptions. Some phonological oppositions as well as phonologically not relevant phonetic facts are not revealed in the written form. Also, some phoneme clusters differ in their orthography from their phonetic form.

Long vowels and consonants are usually reflected in their orthography – they are written as double letters. The most notable exceptions are [Eek and Meister, 1999]:

- Intervocalic plosives /k/, /p/, /t/ are written with *g*, *b*, *d* in the Q1 foot, with *k*, *p*, *t* in the Q2 foot and *kk*, *pp*, *tt* in the Q3 foot, e.g. *tugi*, 'support', nom. sg., *tuki*, 'brand', gen. sg., *tukki*, 'brand', part. sg.;

- At the beginning of the word, a single *k*, *p*, *t* is used, except for some foreign loan words where *g*, *b*, *d* is used at the beginning of the word as in the original language.

- Intraword plosives that neighbour any voiceless consonants are written using a single *k*, *p* or *t* regardless of the foot, e.g. *kopsik* (Q2 foot), *aktus* (Q3 foot). There are some exceptions (e.g. compound words, morpheme boundaries, foreign words) where *g*, *b*, *d* can occur in the neighborhood of a voiceless consonant (e.g. *sead/ma – sead/sin*, *kodakond/ne – kodakond/sete*).

- After long vowels or diphthongs, regardless of the quantity degree of a foot, long (geminate) obstruents (except s, e.g. *poiss*, Q3 'boy', nom. sg.) are written by one letter, e.g. *saate*, Q2 'get', 2.pl.pres., *saate*, Q3 'dispatch', gen. sg., *laat*, Q3 'fair', nom.sg.; to this group belongs also a geminate h in the Q2 foot;

- After a sequence of a short or long vowel (or diphthong) and a sonorant, regardless of the quantity degree of a foot, geminate obstruents (except s, e.g. *varss*, Q3, 'foal', nom. sg.) are written by one character, e.g. *narta*, Q2, 'dogsledge', nom.sg., karta, Q3 'fear', da-infinitive, *kart*, Q3, 'a type of car', nom. sg.;

- Plosive geminates when followed by a sequence of a voiced consonant and a vowel, are also written by one character, e.g. *rütmi*, Q2 'rhythm', gen. sg., *rütmi*, Q3 'rhythm', part. sg.;

Long syllable-final *üü* is pronounced as *üi* (both in a Q2 and Q3 foot) if the following unstressed syllable begins with a short vowel (e.g. *püüa*, Q2 [pyiɑ] 'catch', 2. sg. imperat.).

Long intervocalic /i/ in the Q3 foot is written as *jj* (e.g. *majja*, Q3, 'house', adt. sg.).

Palatalization is not revealed in the orthography (e.g. *palk*, Q3 [pɑlkːk] 'timber', nom.sg. – *palk*, Q3 [pɑlʲkːk] 'salary', nom. sg.).

The phoneme /n/ is realized as palato-velar nasal [ŋ] before palato-velar plosives (except in the case of morpheme boundary before g, k). This fact is not revealed in the orthography (e.g. *istungi* [istuŋki] 'meeting', gen. sg. and *istungi* [istunki] 'I am even sitting').

Foreign loan words are usually written according to the simplified pronunciation rules of the original language. Foreign names are written as in the original language, except for some common placenames that have been adapted to Estonian.

3.3 Morphology and syntax

Typologically, Estonian is in a transition from an agglutinative to a inflected language. Agglutinative languages are characterized by the fact that morphemes carrying grammatical information are appended to word stems, and every such morpheme has only one meaning. In reality, Estonian is rapidly moving away from agglutination and closer to inflection where each morpheme has several grammatical meanings [Sutrop, 2004].

The principal way of forming words in Estonian is by adding derivative affixes to the stem. Estonian has about one hundred derivative affixes, almost all of them are suffixes.

Estonian is a so-called compounding language, i.e. compound words can be formed from shorter particles to express complex concepts as single words. For example, the words *rahva* 'folk' and *muusika* 'music' can be combined to form a word *rahvamuusika* 'folk music' and this in turn can be combined with the word *ansambel* to form *rahvamuusikaansambel* 'folk music group'.

In Estonian language, neither nouns nor pronouns have grammatical gender.

There are no words that consist of only one letter.

Estonian nouns have 14 cases. In comparison: Finnish has 15, Russian has six, German four and English only two cases. The meaning conveyed by case endings in Estonian is expressed by prepositions in English and other languages. By contrast, Estonian has almost no prepositions.

The 14 cases are listed in table 3.6 [Sutrop, 2004]. The case endings are the same in singular and plural, the plural is distinguished by suffixes (*ilusa-lt maja-lt*, abl. sg. – *ilusa–te-lt maja–de-lt*, abl. pl.).

Nouns and adjectives are declined in the same way. If used together with a noun, declination of the adjective 'agrees' with the primary word, except for the last four cases. However, adjective always 'agrees' with the primary word in number ((*ilusa maja-ni*, term. sg. – *ilusa-te maja–de-ni*, term. pl.).

The form of the semantic cases (4th - 14th) can be always derived from the genitive case by adding a suffix. The formation of the genitive and partitive case however depend on the word and is governed by a set of so-called word types. Given a noun and its type, the declensions of the word can be derived from the

'master' word of that type by analogy.

Table 3.6: Estonian cases [Sutrop, 2004].

Case	Example	Meaning
Grammatical cases		
1. Nominative	ilus tüdruk	(a) beautiful girl
2. Genitive	ilusa tüdruku	of a beautiful girl; a beautiful girl (as a total object)
3. Partitive	ilusa-t tüdruku-t	a beautiful girl (as a partial object)
Semantic cases		
Interior local cases		
4. Illative	ilusa-sse maja-sse	into a beautiful house
5. Inessive	ilusa-s maja-s	in a beautiful house
6. Elative	ilusa-st maja-st	from a beautiful house
Exterior local cases		
7. Allative	ilusa-le maja-le	onto a beautiful house
8. Adessive	ilusa-l maja-l	on a beautiful house
9. Ablative	ilusa-lt maja-lt	from on a beautiful house
Other cases		
10. Translative	ilusa-ks tüdruku-ks	[to turn] (in)to a beautiful girl
11. Terminative	ilusa tüdruku-ni	up to a beautiful girl
12. Essive	ilusa tüdruku-na	as a beautiful girl
13. Abessive	ilusa tüdruku-ta	without a beautiful girl
14. Comitative	ilusa tüdruku-ga	with a beautiful girl

Estonian verbs are conjugated in the active and passive voice, and indicative, imperative, conditional and indirect mood, and in the affirmative and negative form. The present, past simple, present perfect and past perfect of the verbs are distinguished. The verb inflections are listed in table 3.7.

Similarly to other Finno-Ugric languages, Estonian uses relatively many postpositions, as opposed to prepositions in languages like English. However, there is a tendency to substitute postpositions for prepositions, and in many situations a word can act both as a preposition and as a postposition (e.g. *mööda teed* vs *teed mööda*, 'along the road' vs 'the road along').

Estonian is known as a free word order language [Remmel, 1963]. Usually many words in a sentence can be easily reordered without the sentence becoming ungrammatical. The reason for this is that the grammatical relations between words are reflected in case endings and inflections. In the literary language, the most common word order is SVO (subject-verb-object). However, the XVS word order (where X stands for any lexical category, capable of forming a phrase) is almost as common in Estonian as SVO [Tael, 1988]. This shows that the

Table 3.7: Estonian verb inflections.

Plurality	Person	Example	Meaning
Singular	I	(ma) armasta-n	'I love'
	II	(sa) armasta-d	'you love (sg.)'
	III	(ta) armasta-b	'he/she/it loves'
Plural	I	(me) armasta-me	'we love'
	II	(te) armasta-te	'you love (pl.)'
	III	(nad) armasta-va-d	'they love'

word order in Estonian is determined by the needs of organizing known and new information rather than by the purely syntactic criteria. The XVS word order is more used in standard language, in spoken language and dialects it occurs only about 12-19% [Lindström, 2000]. A more concise handling of Estonian word order can be found in [Ehala, 2006].

Chapter 4

Acoustic and pronunciation modelling

This chapter presents an approach to modelling Estonian speech sounds in an LVSCR framework that uses hidden Markov models for modelling acoustic units. First, the inventory of acoustic units is proposed. A phonetic classification for building context-sensitive units is given. Next, an alternative approach to selection of acoustic units that takes into account the effect of quantity degrees is proposed. Finally, a simple algorithm that can derive word pronunciations from morphologically tagged word forms is presented.

4.1 Selection of acoustic units

In order to be able to compose a pronunciation dictionary for each vocabulary word or morpheme, one first needs to decide about the inventory of acoustic units. There are many factors that should be considered when making this decision. First, the selected units should be accurate, trainable and generalizable as explained in chapter 2.3.2. Second, since there are no large-scale pronunciation dictionaries for Estonian words, it should be possible to automatically generate a pronunciation for each entry in the system vocabulary.

The most straightforward approach is to use the Estonian phonemes as defined in chapter 3.1 as basic units for acoustic modelling.

There is little difference in the quality of the short and long Estonian phonemes, therefore it is not justified to define separate acoustic units for long phonemes. Instead, long phonemes can be modelled by a sequence of two corresponding short units. The short and long phonemes differ mainly in the duration, and as hidden Markov models are known to model durational characteristics of phonemes poorly, it seems that it is more reasonable to model the long duration by just forcing the model to go through more states. In practice, this is achieved by modelling long durations by a sequence of two models.

However, when using this approach, we lose the ability to have the second part of the long phoneme context dependant on the previous phoneme and the first part dependant on the next phoneme: e.g. in the word *kooli* [k o o lʲ i], the first phoneme is regarded as a context-sensitive triphone k-o+o] and the second [o] as a triphone [o-o+lʲ]; if we would use separate units for long phonemes, the word *kooli* would be modelled as [k oo lʲ i], the long phoneme would be regarded as a triphone [k-oo+l]. This shortcoming should not have a strong negative effect on the modelling accuracy, since the quality of the first part of a long phoneme is not much influenced by the following phoneme, and vice versa. The benefits of having more training samples for estimating the model on a short phoneme due to treating long phonemes as a sequence of two short units should compensate for this problem.

There are 36 segmental diphthongs in Estonian. Diphthongs act similarly to long vowels, therefore it should be practical to apply a similar approach to modelling diphthongs as it was done for long vowels, i.e. diphthongs can be modelled by sequences of corresponding short phone units. Also, the large number of different diphthongs makes it unfeasible to build separate model for each of them, as opposed to English where each diphthong, e.g. [eɪ] as in *day*, is usually modelled by an independent model.

The only exception in handling the short vs. long opposition is the modelling of plosives. The articulation of a plosive requires a closing phase, an obstruction phase and a release phase. The primary difference between short and long plosives is the presence and length of the silent region in the obstruction phase. Thus, it is incorrect to treat a long plosive as a concatenation of two short plosives, and it is justified to create separate units for short and long plosives. In the following, when writing word pronunciation, we refer to a short plosive using a lowercase letter (e.g. *laba* [lapa]) and to a long plosive using an uppercase letter (e.g. *lapi* [laPi]).

It seems to be a good idea to merge pairs of palatalized and unpalatalized phonemes (i.e. [tʲ] and [t], [lʲ] and [l], [sʲ] and [s], [nʲ] and [n]) into single acoustic units. This brings a number of benefits. First, there is no difference in the graphemic representation of the words, if the words differ in only a palatalization of a phoneme (e.g. *palk* [palʲkːk] and *palk* [palkːk]). Second, the palatalized and unpalatalized consonants have only a little difference in their sound and quality, thus such merging increases the training data available for estimating the parameters of the merged unit, making it more robust; the parameters that are sufficiently different for the palatalized and unpalatalized versions should be modelled automatically separately if Gaussian mixtures are used in HMM modelling. Third, it is not easy to determine the correct palatalization in all cases from the word orthography: the Estonian morphological analyser does have the function of marking places of palatalization but the accuracy of this process is not very high. Thus, the training samples of palatalized and unpalatalized phonemes might become too polluted with data that has been incorrectly tagged

as palatalized or unpalatalized.

Handling palatalized phonemes as separate units does have some benefits. If using pronunciation-specific words in the vocabulary (e.g. if *palk* [palʲkːk] and *palk* [palkːk] are separate entries in the vocabulary), the N-gram estimates for such words should become more accurate, given that we are able to assign a correct pronunciation for each word in the language modelling training corpus. For example, the probability of a bigram *kõrge palk[palkːk]* should become much higher than the probability of *kõrge palk[palʲkːk]*. However, such approach would also increase the vocabulary size, as two entries would be needed to represent each word that has two meanings and two corresponding pronunciations, differing only in a palatalization. This has a negative effect on the OOV-rate and thus possibly also on the recognition result, if the vocabulary size is limited only to N most likely words or morphemes.

Using the proposed acoustic inventory, we can assign each of the units into multiple categories in order to automatically classify the HMM senones into context-dependant clusters using a binary tree, as discussed in section 2.3.3. Table 4.1 lists phonemic categories of the proposed units that can be used for generating the decision tree.

4.2 Handling quantity degrees

As discussed in section 3.1.3, Estonian has three distinctive quantity degrees. However, as the quantity degree is a property of a foot rather than a phoneme or a syllable, they can not be modelled using segmental hidden Markov models.

Fortunately, it turns out that for general speech recognition purposes, it is usually not needed to identify the correct quantity degree in order to correctly recognize an orthographical word, as in most cases, there is no difference in the orthography of a word that has the second vs. the third quantity degree. For example, the words *kooli* [kooli] and *kooli* [koo:li] are both written in the same manner. Thus, we may safely ignore the quantity degree difference in most cases. The only difference occurs if the stressed syllable ends in a geminate plosive and is followed by a vowel, e.g. *koti* [kotti] vs. *kotti* [kott:i]. If the words are modelled by the same sequence of acoustic units (i.e. /kotti/), it is not possible to make a distinction between them, using only the score of the acoustic model. However, it may be hoped that the language modelling score for the two words in the given context is different and can be used to make a correct distinction.

The three-way contrast of quantity degrees can only be perceived if information from the next syllable is available. It is even reasonable to say that the overlong quantity degree is not realized by making the stressed syllable longer, but rather by a relative shortening of the vowel of the unstressed second syllable. There is also a certain degree of quality degradation in the realization of the vowel of unstressed second syllable in a overlong foot. Thus, it could be useful to model

Table 4.1: Phonetic categories for Estonian acoustic units.

Question	Units
Consonant	p k t P K T l r m n f v s sh h
Stop	p k t K P T
Short stop	p k t
Long stop	K P T
Stop k	k K
Stop p	p P
Stop t	t T
Dental consonant	t T
Labiodental consonant	f v
Alveolar consonant	t T s n l r
Bilabial consonant	p P m
Glottal-oral consonant	h
Lateral consonant	l
Fricative	f v s sh h
Nasal	m n
Thrill	r
Spirant	v f s sh v r l j h
Sibil	s sh
Liquid	l r
Consonant, front	p P m v f
Consonant, central	t T n s l r
Consonant, back	j sh K k h
Consonant, voiced	l r m n v j
Consonant, voiceless	p P t T s sh f
Vowel	a e i o u ou ae oe ue
Front vowel	e i ae oe ue
Back vowel	a o u ou
High vowel	i u ue
Medium-high vowel	e o oe
Non-low vowel	ou
Low vowel	a ae
Rounded vowel	u ue o oe
Unrounded vowel	i e ae ou a
Semivowel	v j
Voiced	a e i o u ou ae oe ue l r m n v j

such extra short degraded vowels using separate acoustic units $[a^-]$, $[e^-]$, $[i^-]$, $[o^-]$ and $[u^-]$. For the vowels [õ], [ä], [ö] and [õ], such distinction is not needed as they only occur in unstressed syllable of some rare foreign words (e.g. *embrüo* [em:brüo]). However, it is important to note that such modelling approach cannot be very accurate since the most relevant feature in identifying quantity degrees is the duration ratio of the syllables, not the quality degredation and shortness of the vowel of the unstressed syllable. Furtermore, it can be predicted that such short and quality-degraded vowels also occur in other places, e.g. at the end of long words. Because of this, our baseline approach is to ignore quantity degrees. To properly model acoustics of the quantity degrees, proper duration ratios should be taken into account in a general framework.

4.3 Pronunciation dictionary composition

If the distinction between the second and third quantity degrees is completely ignored, and palatalized and unpalatalized variants of consonants are merged into one unit, the pronunciation dictionary can be drawn almost directly from the morphologically tagged word orthography. Morphological tagging is needed because borders between compound word particles have an important effect on the way plosives are pronounced (e.g. in words *elu+tuba* [elutuba] *vs elutu* [eluTu], the middle *t* is pronounced as [d̥] as in *tuba* if there is a border between compound word particles in front of it, or as [t] if the *t* is in the middle of a simple word).

The following letter-to-phoneme transformations are needed for the pronunciation dictionary composition. The order of transformations is important, as multiple transformations could be applied to a certain context (e.g. *aqua* is transformed to [akva] and later to [aKva]).

- *c* → *ts*, if *c* is followed by *o* or *e* (e.g. *cicero* [tsitsero]), otherwise *c* → *k* (e.g. *curriculum* [kurriKulum]
- *w* → *v* (e.g. *wiiralt* [viiralT]
- *y* → *i* (e.g. *kelly* [kelli]
- *qu* → *kv* (e.g. *aqua* [aKva]
- *zz* → *ts* (e.g. *pizza* [pitsa])
- (*kk,k*) → *K*, (*pp,p*) → *P*, (*t,tt*) → *T* , if preceded by a voiced phoneme and followed by a word end, compound word particle border, or a voiced phoneme (e.g. *kapp* [kaP], *kapi* [kaPi], *kappi* [kaPi], *karpi* [karPi], *karp+kala* [karPkala], but *kapsa* [kapsa], *kast* [kast])
- *g* → *k*, *b* → *p*, *d* → *t* (e.g. *kabi* [kapi], *banaan* [panaan])

- *üüi* → *üi*, if followed by a vowel (e.g. *müüa* [müia])
- *z* → *s* (e.g. *zoo+park* [sooparK])
- *ž* → *š* (e.g. garaaž [karaaš])

Chapter 5

Language modelling for large vocabulary recognition

This chapter presents a range of techniques that try to solve the most serious problem in the development of Estonian LVCSR – efficient and robust statistical language modelling. The approach relies on Estonian automatic morphological analysis and the treatment of morphemes as basic language units.

In the next sections, the motivations for modelling language using sub-word units is first examined, followed by an overview of several techniques that have been used for modelling other inflective languages. A linguistically motivated method for Estonian that uses units derived through morphological analysis is the presented. Next, statistical approaches for selecting language model vocabulary and estimating sub-word N-gram probabilities from union of training corpora are presented. A method for reconstructing compound words from the pseudo-morpheme based decoder output, using only statistically derived linguistic knowledge, is then developed. Finally, the last section introduces two independent techniques that improve the robustness of sub-word based N-gram models, given a sparse training corpus.

5.1 Pseudo-morpheme based language modelling

Estonian is a highly-inflecting language and has thus a large number of inflected forms for each word-phrase. For speech recognition systems, an inflected form is considered as a different word. This is due to the fact that inflected words' pronunciation is different from the base form, and have different syntactic roles and usage patterns. In addition, new compound words can be constructed on the fly by using concatenation, and new verbs and adjectives can be derived from nouns by adding suffixes. This results in a practically unlimited number of total distinct possible words in the language which creates a number of challenges in constructing a language model for large vocabulary speech recognition system.

The first problem lies in data sparsity. In order to recognize a word, a conventional word-based language model must have this word in its vocabulary. However, it is not feasible to enter all possible inflections, compound and derived words into the vocabulary, as this would create an extremely large word list and it would still be impossible to predict all compound and derived word formations. Also, it would be computationally and spacially very expensive to compute and store N-gram probabilities for all observed word combinations. The second, and more important problem lies in the fact that most words in their all but most frequently used inflections are only rarely seen in training data. This makes it very difficult to robustly estimate their N-gram probabilities.

A common solution to the language modelling problem described above is to split some words into smaller, "subword" units. The smaller units can be more adequately modelled and after recognition, they can be used to reconstruct the original words. The subword units decrease the vocabulary size as the number of distinct units is much less than the number of distinct words. The probabilities of the smaller units can be also much more robustly estimated from training corpora, as they occur more frequently than all the inflected words that can occur in the language.

In the following subsections, an overview of the existing approaches to the subword-based language modelling is given first. Then, a method that relies on Estonian morphological analyser for splitting the words into shorter units is described.

5.1.1 Related work

There are a few different methods to find the appropriate splitting for each word. The most common approach uses full morphological analysis and disambiguation to decompose words into morphemes. The morphological analyser has usually a list of known stems, endings and other morphemes and uses a set of rules to find the correct decomposition for a word. As many words have multiple possible decompositions, a disambiguator is usually applied that selects the most probable classification based on the surrounding context. Disambiguator is commonly implemented using hidden Markov models or some other statistical classifier. After decomposition, the most frequent morphemes are used as the vocabulary of the recognition system. This approach has been used for Japanese [Ohtsuki et al., 1999], Korean [Kwon and Park, 2003], Hungarian [Szarvas and Furui, 2003] and other languages.

An alternative way to split words into smaller units is to use a data-driven algorithm instead of a morphological analyser. The data-driven algorithms are usually language independent and can be applied to all inflected languages.

In [Maucec et al., 2003], a data-driven method is proposed that is applied to Slovenian language. The method assumes that a word consists of a stem and an optional ending. The algorithm first collects all common word endings from the

training corpora and then decomposes the words using a longest match principle. In order to avoid over-stemming, the decomposition uses a restriction that the stem must be of predefined minimum length. An empty ending is added if a word cannot be decomposed. In the language model, the computation of the probability of a stem is based on the knowledge of two preceding stems, and the probability of an ending is computed from the knowledge of two preceding endings.

In [Siivola et al., 2003, Creutz and Lagus, 2005] a data-driven algorithm called Morfessor is presented that is able to find morpheme-like units called statistical morphs from a large text corpus. The optimal units are found according to a cost function that is based on the minimum description length (MDL) principle. The minimized cost function is the coding length of the lexicon and the words in the corpus represented by the units in the lexicon. The algorithm tends to give units that are both frequent and as long as possible to suit well for both training language models and speech recognition. This method guarantees full coverage of the language by splitting rare words into very short units, if needed. Recently, this method was also applied for Estonian, using the SpeechDat speech corpus for training and testing [Hirsimäki et al., 2006]. Using a morph-based language model in a LVCSR task resulted in a WER of 47.6%, compared to 56.3% when using a conventional word-based language model.

There are some different approaches for dealing with the problem of vocabulary growth in LVCSR. In [Geutner et al., 1998], a two-pass recognition approach is presented, where the first pass uses a word-based vocabulary to derive a lattice of potential words. The list of all words in the lattice is augmented by the most likely words (in terms of number of observations in a huge text corpus) which are acoustically similar to the words observed in the lattice. A second recognition run is then carried out using the adapted vocabulary. An absolute improvement of 5.8% is reported on a Serbo-Croatian recognition task. Another alternative approach advocates the use of huge vocabularies for inflected languages. In [McTait and Adda-Decker, 2003], the use of a lexicon of 300 000 instead of 60 000 words lowered the word error rate from 20.4% to 18.5% for a German LVCSR task. Similarly, [Nouza et al., 2005] use a 312 000 word lexicon for Czech broadcast news transcription and achieve a word error rate of 18.4%.

5.1.2 Decomposition of words using morphological analysis

The approach presented in this thesis relies on Estonian morphological analyser [Kaalep, 1998b] and disambiguator [Kaalep, 1998a]. Using morphological analysis as the basis for language modelling has some advantages over language-independent data-driven methods. Namely, the analyser tags the boundaries between compound word particles and inflectional endings. This information is useful in later phases of language modelling and in pronunciation dictionary composition. It also provides part-of-speech tag for each word in the sentence. The part-of-speech tags are later used for improving vocabulary selection (see section

5.2).

The Estonian morphological analyser is made up of three logical parts: a sentence splitter, a morphological analyser and morphological disambiguator. The sentence splitter marks sentence boundaries in written text. The morphological analyser decomposes compound words and marks the boundaries between stems and suffixes. The tool is implemented so that the words in the running text are compared with the combinations of lexemes in the dictionary. No two-level rules are applied to the comparison. The textual words are analysed from the right to the left, i.e. the endings and suffixes are cut off, and the base(s) are checked with the help of the lexicon, which contains the stems of 38,000 words (67,000 items in total as many words have different stems depending on the inflection). The splitting process is controlled by a complex ruleset that defines which lexemes can be concatenated. A rule-based guesser is applied for words that cannot be analysed using the dictionary. Such words constitute up to 3% of all words in an average text. The morphological disambiguator tries to select the correct analysis for words that have multiple analyses (around 40% of all words), using words' local context. The disambiguator uses a statistical approach and is based on hidden Markov models. According to [Kaalep and Vaino, 2001], about 3% of all analysed words get a wrong analysis because of disambiguation.

Before splitting the analysed words into morphemes, they are processed by a tool that attaches pronunciations to them. The pronunciation generation is done before splitting because in some cases, the pronunciation of a morpheme depends on the preceding morpheme. The pronunciation generation tool retains morpheme boundaries in words, so that later, morphemes can be matched with their corresponding pronunciations. As a result, there are orthographically similar morphemes in the training data that have different pronunciation, depending on their context. During splitting, all endings and suffixes used for word derivation and inflection are specially tagged (using an underscore). This makes it possible to attach the endings back to the previous stem after decoding. As one-phoneme words are acoustically very confusable and have been shown to yield many recognition errors [Kwon and Park, 2003], the one-letter suffixes are not split from the previous morphemes. Compound word boundaries are not retained when training the morpheme language model. For reconstructing compound words from the decoder output, a separate probabilistic hidden event language model is used (see section 5.4). Some examples of the splitting process are given in table 5.1. Compound word boundaries are marked with '+' and endings are separated from the preceding morpheme by '_'. Note how the ending *_te* has two pronunciations, depending on the context, and that the word *nei_d* is not split due to the one-letter suffix rule.

The smoothed N-gram language model assigns at least a small likelihood to any input sequence. Thus, the language model accepts also invalid sequences of pronunciation-specific morphemes. For example, the language model accepts a sequence *küsimus[k ü s i m u s] _te[tt e]* which shouldn't be allowed, as the long

Table 5.1: The splitting of words into pronunciation-specific pseudo-morphemes.

Original word	Morphological analysis	Pronunciation	Splitting
küsimuste	küsimus_te	k ü s i m u s _ t e	küsimus[k ü s i m u s] _te[t e]
saadete	saade_te	s a a t e _ tt e	saade[s a a t e] _te[tt e]
neid	nei_d	n e i _ t	nei_d[n e i t]
ajuvaba	aju+vaba	a j u + v a p a	aju[a j u] vaba[v a p a]

plosive *tt* in the beginning of suffix *_te* cannot occur after *s*. Another problem with the morpheme-based language model is that it accepts morpheme sequences that can be pronounced but are grammatically invalid, such as *küsimus[k ü s i m u s] _id[i t]* which produces an invalid word *küsimusid*. There are a few ways to tackle this problem. In [Szarvas and Furui, 2003], the morphosyntactic grammar that defines the permitted morpheme combinations is built directly into the language model by means of weighted finite state transducer (WFST). The resulting stochastic morphosyntactic language model is an intersection of the stochastic N-gram language model and a morphosyntactic grammar and eliminates the invalid combinations from the language model while retaining the likelihoods of the valid transitions. However, the drawback of this method is the increase in the complexity and size of the language model: around 2.5-fold increase in the number of arcs of the trigram language model is reported. Another way to reduce the errors resulting from combining invalid morpheme sequences is to generate *N*-best lists for each sentence and select the hypothesis that contains the smallest number of morphosyntactic errors. The drawback of this method is that it is not guaranteed that a good candidate would be included in the list for reasonable values of *N*.

The baseline approach presented here does not deal with the problem of invalid morpheme sequences. By analyzing the recognition errors in the experiments it was observed that illegal morpheme combinations consitute only a very small part of the errors. Apparently a stochastic N-gram trained on a reasonable amount of data almost eliminates such problems. Additionally, given the 2.5-fold increase in the complexity of the stochastic morphosyntactic language model, it might be more reasonable to use a higher order N-gram or a wider search beam instead, if processing power is sufficient and a higher recognition accuracy is needed.

The method presented later in this thesis in section 5.5.2 tries to tackle some of the problems resulting from using morphemes as basic units in a standard stochastic N-gram language model.

5.2 Vocabulary selection

The vocabulary of a continuous speech recognition system is one of the key factors in determining its performance. A speech recognition system can only recognize words (or morphemes in our case) that are in it's vocabulary, thus ideally, the vocabulary should be large and comprehensive so that it can cover any sentence in input speech. If a word in the input speech is out-of-vocabulary (OOV), an acoustically and contextually similar word is recognized. Often, the OOV word is replaced with many shorter words, or the OOV-word and some of its surrounding word is replaced with one similar long word. The replacement error also confuses the language modelling context, resulting in potentially more errors in recognizing the surrounding words. Many investigations report an average of 1.2 [Rosenfeld, 1995] to 1.6 [Woodland et al., 1995] recognition errors which would not occur, had the OOV-word been in the vocabulary.

While a large vocabulary may be good for lexical coverage, it turns out that it is profitable to settle for smaller and more tractable vocabularies. First, the language models built on large vocabularies are very large. Large language models require a lot of processing and memory resources during recognition. Large vocabularies contain a lot of rarely seen words, and it is not possible to robustly estimate their language model probabilities in various contexts. And finally, large vocabularies contain many acoustically confusable words, which usually results in many word substitution errors.

Thus, it is very important to find a good balance between two sources for word recognitions errors – the OOV words and the large language model. Usually, a number of text corpora from various sources and time periods are available. Given a collection of training corpora, the most straightforward approach for picking the vocabulary for large vocabulary recognition is to order all words in the corpus by frequency and pick the predefined number of words from the top of the list. This problem can be regarded as estimating unigram probabilities of the test distribution, and ordering the words by theses estimates. However, this simple approach has same caveats: if the training data is not balanced, the vocabulary could become very biased towards the type of data that is the most prevalent. For example, given a 200-million word corpus of legal texts and a 20-million corpus of prose texts, it is clear that a simple mixture of the corpora is strongly biased towards legal terms. A simple solution is to normalize word counts in different corpora using the corpora size. Given that a word w_i occurs in corpus j $n_{i,j}$ times, and the number of words in a corpus j is N_j, the normalized total count Φ_i of a word w_i can be written as

$$\Phi_i = \sum_j \frac{n_{i,j}}{N_j}$$

Given some knowledge about the recognition task, one might also manually

assign weights λ_j to different types of corpora j, depending on the relevance of the corpus to the target domain, and renormalize the normalized word counts using the weights. The renormalized counts are then given by

$$\Phi_i = \sum_j \frac{\lambda_j n_{i,j}}{N_j}$$

The manual estimation of the weights λ_j is clearly not the most optimal solution. If a sample text of a target domain is available, the weights can be estimated automatically, as proposed in [Venkataraman and Wang, 2003]. The idea is based on the assumption that the vocabulary of the sample text is related to the vocabularies of each training corpus, and the full vocabulary of target domain can be inferred from the individual training corpus vocabularies, considering the observable portion of the domain text to be a sample. The weights λ_j can be then estimated using the maximum likelihood principle: we simply interpret the normalized counts $\frac{n_{i,j}}{N_j}$ as probability estimates of a word w_i given corpus j and the weights λ_j as mixture coefficients for linear interpolation. The weights λ_j must be chosen so that the probability of the sample text vocabulary is maximized. Formally, let $P(w_j|j) = \frac{n_{i,j}}{N_j}$. The goal is to find

$$\hat{\lambda_1}, \ldots, \hat{\lambda_m} = \text{argmax} \prod_{i=1}^{|V|} (\sum_j \lambda_j P(w_j|j))^{C(w_i)}$$

where count $C(w_i)$ is the count of w_i in the sample text and V is the set of words in the vocabulary. The weights λ_j can be estimated using the EM algorithm.

The largest part of language model training corpora consists often of newspaper texts. Newspaper texts have some characteristics that are not desirable from language modelling point of view: they contain many proper names (both native and foreign), many acronyms, abbreviations and numerals, such as year and date ordinals. The abundance of proper names creates two kinds of problems: first, many of the names are of foreign origin, and the automatic Estonian pronunciation rules do not handle them correctly. As a result, they would be put to the vocabulary using the wrong pronunciation, making their correct recognition very improbable. Second, the sheer amount of often occurring names may make the vocabulary overpopulated with proper names, and leave too little room for other words. This is of course arguable, and depends a lot on the task: for broadcast news domain, where proper names are as common as in newspaper texts, the big amount of proper names in the vocabulary might be favorable. There may even be a strategy to give a bigger weight to names that occur often in recent texts. The pronunciation problem might be handled using a large hand-crafted name pronunciation vocabulary or some data-driven language-independent grapheme-to-phoneme conversion algorithm (such as in [Bellegarda, 2005]). For abbreviations

and acronyms, there are two reasonable approaches: two have them expanded before language modelling training and vocabulary selection, or to handle them as separate words but add their correct expanded pronunciation to the the dictionary. The latter does not work for Estonian, as the abbreviation itself can be expanded in many ways, depending on its inflection in the current context. A similar approach can be taken for numerals written as figures (such as year numbers). For both abbreviations and numbers, the normalization into a readable full-length form with the correct inflection is not a trivial task. Many data-driven algorithms (such as in [Kanis et al., 2005]) that can deal with this problem have been proposed.

In this work, a rather simplified approach is taken to tackle the problem of text normalization. The speech recognition test data used in the experiments is known to be quite different from newspaper texts and contain few proper names, numerals and no abbreviations. Thus, all morphemes containing numerals and all morphemes that are parts of words tagged as abbreviations are filtered out before selecting the vocabulary for the language model. Proper names are handled as follows: at first, all words tagged as proper names are removed from the candidate vocabulary; next, a fixed number (1% of the language model vocabulary size) of most frequent morphemes that are parts of proper names are readded to the vocabulary candidate list. Of course, this does not mean that 1% of the final vocabulary consists of unique proper names: many of the proper names have multiple different stems (such as *tallinn*, *tallinna*) that occur frequently enough to be added to the vocabulary. Section 6.2.2 contains a detailed analysis of experimental results using this method.

5.3 Training a morpheme trigram language model

After fixing the language model vocabulary, a statistical language model can be estimated using the training corpora. The simplest approach would be to concatenate all the training corpora and compute the N-gram probabilities from the combination of all counts. However, this can have some drawbacks if the training data is unbalanced, just like when selecting language model vocabulary. Therefore, it is usually more profitable to train a separate N-gram model for each domain/corpus and finally combine the models linearly. Given k models $P_i(w|h)_{i=1...k}$, the combined model is defined as:

$$P_{combined}(w|h) = \sum_{i=1}^{k} \lambda_i P_i(w|h)$$

where $0 < \lambda_i <= 1$ and $\sum_i \lambda_i = 1$.

The weights λ_i can be assigned manually, or estimated automatically using a sample of in-domain text using the Estimation-Maximization (EM) algorithm (see [Jelinek, 1989] for details). In the latter case, the resulting model will have

the minimum perplexity possible for the sample text using the mixture of the domain-specific models. If the sample text is large and representative enough, the weights will be nearly optimal for the test data as well.

Using linear interpolation with automatic weight optimization has many advantages over estimating one model from the combination of the corpora. Most importantly, it removes the worries that some certain domain may be over-representative in the training data. Using automatic weight tuning, the domain with a lot of training data will become more accurately modelled, in contrast to being over-represented, as it would be the case with using the copora combination. It is however important to have a reasonable amount of data for each domain and not to have the domains too granually defined – this would create a situation where a domain is not estimated robustly enough and might have a negative effect on the overall performance, if the sample in-domain text is not large and representative enough.

5.4 Reconstructing compound words using a hidden-event language model

The output of the morpheme-based decoder is a sequence of morphemes for each utterance. The set of different suffix morphemes is rather small and thus the suffixes can be tagged in the vocabulary so that they can be concatenated to the previous stem after decoding. However, this approach can not be applied for reconstructing compound words: the set of stems and morphemes that take part in forming compound words is very large and sparse, thus treating part-of-compound morphemes as separate units for language modelling would make the vocabulary very large and it would be impossible to estimate the N-gram probabilities of the compound-forming morphemes with enough robustness.

To overcome the problem of compound word reconstruction we can model compound word connectors as hidden events in the language model. Such language model is typically used for sentence segmentation of conversational speech based on recognized words [Stolcke et al., 1998], but can be generalized for detecting other hidden events between recognized units. The event language model describes the joint distribution of words and events, $P_{LM}(W, E)$. The words and events are treated as a single token stream. For training such a hidden event LM, the hidden events in the training texts must be represented by an additional token, for example:

```
mitme _te taeva <CC> enne _te najal võid
ala _tes teisi <CC> päeva _st...
```

The event "<CC>" is an additional token in the vocabulary that is inserted in the word sequence for LM training.

During event detection, such model can be used as a hidden Markov model in which the word/event pairs correspond to states and the words to observations. The transition probabilities are given by the hidden event N-gram model. Given a word/morpheme sequence, a forward-backward dynamic programming algorithm [Rabiner and Juang, 1986] can be used to find the posterior probability $P_{LM}(E_i|W)$ of event E_i at position i. For our compound connector detection task, we choose the event sequence $\hat{E}$ that maximizes the given posterior probability at each individual morpheme boundary. This approach minimizes the expected per-boundary classification error rate.

It is important that the word vocabulary of the hidden event model is fixed, i.e. the observations should not contain any tokens that are not present in the hidden event model. Fortunately, this is easy to achieve as the output of the decoder contains only the words/morphemes that are in the language model vocabulary. The same vocabulary can be used as the basis of the hidden event model.

After applying this approach, the most probable compound word connectors between recognized morphemes are annotated. The resulting morpheme and compound word connector list can be used to fully reconstruct the morphemes into words, as shown in table 5.2.

Table 5.2: Word reconstruction from (perfect) decoder output and hidden compound word boundary detection.

Phase	Result
Decoder output	mitme _te taeva enne _te najal võid ala _tes teisi päeva _st pika aja _lis _te võimalus _tega kohane _da
Compound word boundary detection	mitme _te taeva <CC> enne _te najal võid ala _tes teisi <CC> päeva _st pika <CC> aja _lis _te võimalus _tega kohane _da
Reconstruction	mitmete taevaennete najal võid alates teisipäevast pikajaliste võimalustega koheneda

The approach described above is very simple but has a few weaknesses. First, it only looks at the local context while the decision on weather the words should be treated as separate units or written together is often based on a much wider context and may sometimes require a full understanding of the discourse. E.g. in the sentence *Kooli(-)õpetajad on targad* 'School teachers are smart / Teachers of the school are smart' the word *kooliõpetajad* 'school teachers' is written as a compound word if one talks about school teachers in general, and separately (*kooli õpetajad*) if one talks about teachers of a certain school.

Another weakness of the method is that it doesn't take into account the prosodic features that are often of high importance in deciding weather words form a compound word or not. For example, usually (but not always) there

seems to be a noticeable word-level stress on the second word of the word pair, if the words are to be written separately. Also, there is sometimes a slightly longer pause between words if they are to be written separately. Therefore, one might try to apply a similar approach that has been used for sentence boundary detection ([Shriberg et al., 1997, Mast et al., 1996]) where CART-style decision trees predict event classes from local prosodic properties at the word boundary of interest. The prosodic features of interest include duration (of pauses, final vowel and final rhymes), pitch (F0 patterns, preceding the boundary, across the boundary) and energy. The prosody-based model might be combined with the language-based model to achieve a better accuracy.

Compound word detection errors are probably not very annoying from users' perspective in tasks like dictations, but have a high impact when measuring word error rate of the recognizer – each compounding error introduces a substitution error and an insertion or deletion error, depending on whether a compound word was mistakenly replaced with two separate words or two separate words were mistakenly compounded, respectively.

5.5 Improving the language model

Modelling Estonian morpheme sequences with conventional N-gram models has some obvious weaknesses. First, the relatively free word order in Estonian means that the variety of word combinations is higher than in languages like English. Given the relatively small corpus size, this makes it difficult to estimate the probabilities of most morpheme N-grams with high robustness. Second, the use of morphemes as basic units in the language model reduces the span of the language model. For example, the sentence "*poisid mängivad jalgpalli*", "boys play football" would be broken into morphemes "*poisid mängi _vad jalg palli*" and the probability of the last particle "*palli*" would be conditioned on only the last two preceding particles "*_vad jalg*", whereas in English, the preceding words "boys play" would be used when calculating the conditional probability of the word "football".

In this section, two methods that are motivated from the described problems are introduced. The first method uses corpus statistics to assign all morphemes of the vocabulary to clusters and thereby make morpheme probability estimates more robust. The second method uses a two-pass approach to apply a word-level N-gram model combined with morphological analysis in the second pass, in order to make the language model span longer and try to eliminate short-term morphosyntactical inconsistencies in recognizer output.

5.5.1 Using statistically derived morpheme classes

Experiments have shown that in order to achieve a good coverage of Estonian sentences, a very large vocabulary is needed even when morphemes are used as

basic units for language modelling. At the same time, the size of the corpora that can be used for training an N-gram language model is more limited than for languages where more written resources are available and more systematic work has been done in collecting the corpora. Additionally, the word order in Estonian is relatively free which causes a high variation in the way words are permuted in the sentence. All those reasons mean that the number of possible grammatically valid N-grams in the language is very high. As a result, a large part of the higher order N-grams that occur in any unseen heldout text are never observed in the training texts, and many of those that are observed, occur only once or a few times, which makes it very difficult to estimate their probability with enough robustness and reliability.

One method that is designed to handle this data sparsity problem is to define word classes that exhibit similar semantic and/or grammatical behavior. For example, it would not be surprising if the probability distribution of words in the vicinity of the word *seitse* 'seven' is very similar to that of the word *kaheksa*, 'eight'. Of course, the distributions would not be identical: there won't be many sentences like *Nädalas on kaheksa päeva*, 'A week has eight days' or word pairs like *kaheksa samuraid*, 'Eight samurais'. Still, if we combine the histories preceding to *seitse*, *kaheksa* and other digits, it may be possible to make more robust predictions for histories that we haven't observed by assuming that the contexts that the digits occur in are similar.

N-gram language modelling using word classes

The principle of class-based language models is to use some component that uses word equivalence classes to capture dependencies in training text. If we assume that a word can be uniquely mapped to only one class, we have a deterministic class mapping function of the form

$$C : w \rightarrow C(w)$$

The class N-gram model can be then computed as follows:

$$\begin{aligned} &P(w_i|C(w_{i-N+1}) \dots C(w_{i-1})) \\ &= P(w_i|C(w_i)) \cdot P(C(w_i)|C(w_{i-N+1}), \dots, C(w_{i-1})) \end{aligned}$$

where $P(w_i|C(w_i))$ denotes the probability of a word w_i given class C_i in the current position (also known as the unigram class membership component), and $P(C(w_i)|C(w_{i-N+1}), \dots, C(w_{i-1}))$ denotes the probability of class C_i given the class history (also known as class N-gram component).

If we have the mapping function defined, it is easy to compute the class membership probability from training text using the empirical frequency of the

word $N(w_i)$ and the class $N(C(w_i))$:

$$P(w_i|C(w_i)) = \frac{N(w_i)}{N(C(w_i))}$$

The class N-gram component can be computed from training texts using the same smoothing and backoff methods as for conventional word N-grams.

Finding clusters

There are several ways to assign words into classes based on syntactic and semantic information that exists for the language and the task. If we have domain knowledge about the task, it is often profitable to cluster together words that have a similar semantic functional role. For example, if we have to develop a conversational information system for city bus travel, we can group the names of bus stops such as *Vabaduse väljak*, *Nõmme*, into one broad class, and city districts like *Mustamäe*, *Lasnamäe* and *Pelgulinn* into another class. Such grouping is particularly advantageous in fighting with the data sparsity problem: the resources for training a language model for such task are always very limited, and otherwise it would be impossible to find a robust probability estimate for a sentence like *Millal väljub viimane buss kesklinna poole Tihase peatusest?*, 'When does the last bus for city center leave from the Tihase stop?'. Given the clustering as explained above, we need to estimate the probability of the sentence *Millal väljub viimane buss [linnaosa, sg. gen] poole [peatuse nimi, sg. gen.] peatusest?*, 'When does the last bus for [district] leave from the [bus stop name] stop?'. This approach makes it also easier to add new names (e.g. new bus stop names) by just adding the name into the corresponding class and using some smoothing method to assign a non-zero probability to the corresponding class membership function. Thus, the new name inherits all the possible word trigram relationships of the class.

For general large vocabulary speech recognition applications, it is impractical to derive word clusters in the same manner as for narrow domain-specific tasks. Instead, data-driven algorithms are used to find word classes which are deemed to be similar in some way. All such algorithms are in fact search procedures to find a class label for each word with a predefined objective function. The majority of the methods try to maximize the log-likelihood of a bigram class model LL_{bi} on the training data by making iterative controlled changes to an initial class function. The log-likelihood of a bigram class is calculated as

$$LL_{bi}(C) = \sum_{i=1}^{N_W} \log \frac{N(w_i)}{N(C(w_i))} \cdot \frac{N(C(w_{i-1}), C(w_i))}{N(C(w_{i-1}))}$$

where N_W is the number of words in the training data. Maximizing the likelihood of the bigram class model is equivalent to minimizing the perplexity of the model against the training corpus.

One such clustering method is the word exchange algorithm [Kneser and Ney, 1993]. As the name suggest, the algorithm exchanges words between the N_C classes until the optimization criterion converges. Each word is moved to a class so as to maximize the increase in the bigram log-likelihood. The algorithm can get stuck in a local maximum and is therefore not guaranteed to find the optimal class function for the training text. In practice, a few extra iterations can compensate for this.

5.5.2 Rescoring using morphological analysis and a factored language model

The usage of sub-word units in a conventional N-gram language model reduces the span of the language model, when compared to a language model that uses words as basic units. The objective of the proposed method is to combine the advantages of particle-based and word based language models using a two-pass approach. The particle-based language model is used in the first pass, granting a good vocabulary coverage. In the second pass, a dynamically constructed word-based language model is used that assigns a more accurate language model score to each sentence hypothesis from the first pass. In addition, the language model applied in the second pass makes use of morphological part-of-speech tags of the words to make the probability estimates more robust and reduce local morphosyntactic errors in recognizer output.

Factored language models

Factored language models (FLM) [Kirchhoff et al., 2002] are used to explicitly represent interdependencies among the morphological components of words both across time and within a word. In an FLM, words are viewed as vectors of k factors, such that $w_i \equiv \{f_i^1, f_i^2, ..., f_i^K\}$. In general, factors can be any features relevant to the word, e.g. part-of-speech tags, word roots, stems, or data-driven word classes or semantic features. A word sequence if length N can thus be converted to K parallel sequences of factors, denoted as $f_1^{1:K}, f_2^{1:K}, .., f_N^{1:K}$. An FLM is a statistical trigram model over factors and can be factored as follows:

$$\begin{aligned} P(f_1^{1:K}, f_2^{1:K}, ..., f_N^{1:K}) &= \prod_{i=1}^{N} P(f_i^{1:K} | f_{i-1}^{1:K}, f_{i-2}^{1:K}) \\ &= \prod_{i=1}^{N} P(f_i^{1:K} | f_{i-1}^{1}, f_{i-1}^{2}, ..., f_{i-1}^{K}, f_{i-2}^{1}, ..., f_{i-2}^{K}) \end{aligned}$$

The factored word representation can be useful during language model backoff, in order to estimate word N-gram probabilities more robustly. In a standard Katz

backoff scheme [Katz, 1987], trigram probability is estimated as

$$P_{BO}(w_i|w_{i-2},w_{i-1}) = \begin{cases} d_{C(w_{i-2},w_{i-1},w_i)}P_{ML}(w_i|w_{i-2},w_{i-1}) & \text{if } C(w_{i-2},w_{i-1},w_i) \geq \tau_3 \\ \alpha(w_{i-2},w_{i-1})P_{BO}(w_i|w_{i-1}) & \text{otherwise} \end{cases}$$

where P_{ML} denotes a maximum likelihood estimate, $C(w_{i-2},w_{i-1},w_i)$ denotes the count of the triple w_{i-2},w_{i-1},w_i, τ_3 is a count threshold, $d_{C(w_{i-2},w_{i-1},w_i)}$ is a discounting factor (generally between 0 and 1) and $\alpha(w_{i-2},w_{i-1})$ is a normalization factor that ensures that the distribution sums to 1. The idea behind Katz backoff scheme is to avoid zero probabilities for unseen trigrams, by backing off to the next lower-order probability distribution. In an FLM, where temporally synchronous as well as temporally successive elements are present, a more flexible order of backing off can be defined. With FLM, the notion of *backoff graph* is introduced. Figure 5.1 depicts a graph of all possible backoff paths for a word, given a history of two previous words, when a word consists of two factors f^1, f^2. Back-off graph defines the order in which the conditioning factors are dropped. The order can be chosen based linguistic knowledge (e.g. always drop more distant and more general factors first), or chosen at run time based on some statistical criteria. Furthermore, multiple backoff paths can be chosen in parallel and their probability estimates can be combined using some non-negative function, such as mean, product, or maximum.

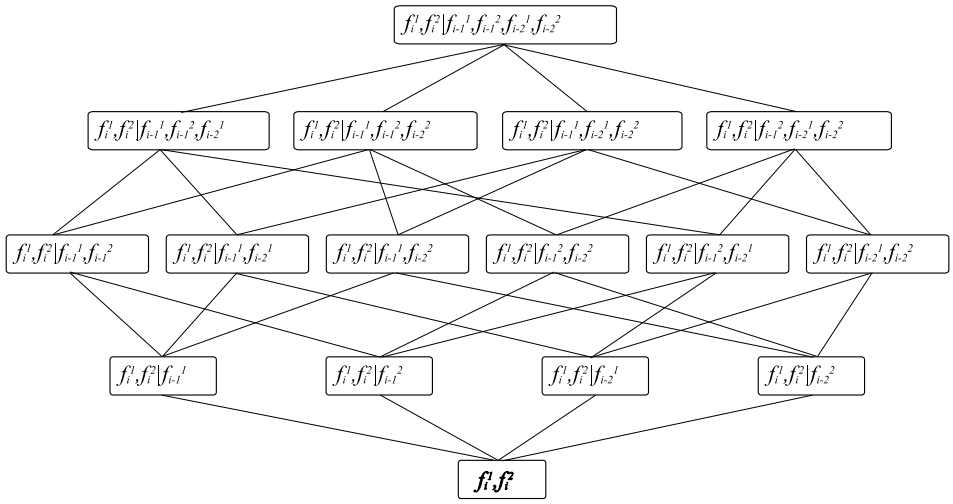

Figure 5.1: All possible backoff paths for a word with two factors, given a history of two previous factor vectors.

The generalized parallel backoff method is implemented by a new backoff function

$$P_{GBO} = \begin{cases} d_c P_{ML}(f|f_1,...,f_L), & \text{if } c > \tau_L \\ \alpha(f_1,...,f_L)g(f,f_1,...,f_L), & \text{otherwise} \end{cases}$$

where c is the count of $f, f_1, ..., f_L$, $P_{ML}(f|f_1, ..., f_L)$ is the maximum likelihood distribution, τ_L is the count threshold, and $\alpha(f_1, ..., f_L)$ is the normalization factor. The function $g(f, f_1, ..., f_L)$ determines the backoff strategy. Several different g functions can be used, including the mean, weighted mean, product, and maximum of the smoothed probability distributions over all subsets of the conditioning factors.

Factored language models have been successfully used for various language modelling and speech processing tasks [Bilmes and Kirchhoff, 2003, Parandekar and Kirchhoff, 2003, Kirchhoff et al., 2006].

System architecture

The architecture of the proposed recognition system is shown on figure 5.2. A decoder using a language model of subword units is used in the first pass. It outputs an N-best list of sentence hypotheses for each sentence, together with their acoustic and language model scores. Each hypothesis is originally a sequence of subword units.

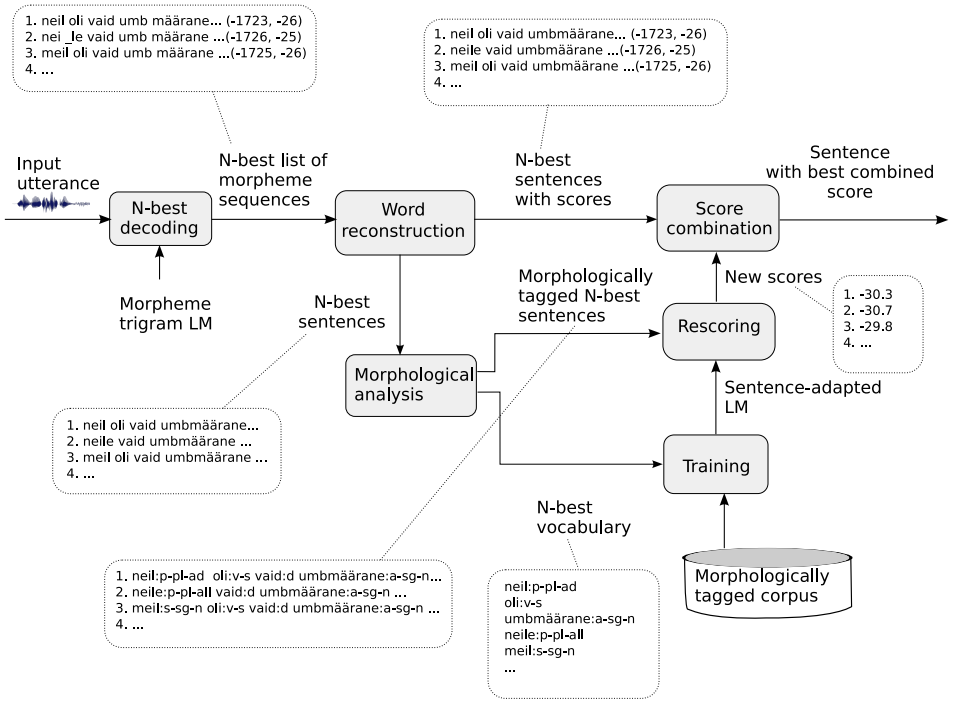

Figure 5.2: Architecture of the two-pass recognizer.

After recognition, all N-best candidates are reconstructed to word sequences. In our approach, this is done by concatenating suffix particles back to the preceding stems and reconstructing compound words using a separate hidden event

language model.

In the next phase, the N-best sentence hypothesis are processed by a morphological analyser and disambiguator, that attaches part-of-speech (POS) tags and stem information to each word form.

The morphologically tagged sentence hypotheses are used to create a vocabulary for the dynamic sentence language model. The sentence language model is estimated from morphologically tagged training corpora. To speed up the computation of the model, the N-gram counts of all word forms may be precomputed. The resulting language model will only contain the probabilities that are needed for estimating the scores for the sentence hypotheses for the current sentence, thus the vocabulary is fixed and small, and the OOV rate is effectively zero. However, many of the word form sequences in the N-best list are usually never or only very rarely seen in the training corpora. Therefore, we propose the using of a factored language model (FLM) as the dynamic sentence model. In our case, the factors are the word itself, its POS tag, and its stem. In a factored model, a word probability can be estimated based on the preceding POS-tags and/or stems, whenever there is insufficient data to fully estimate the probability based on the preceding words.

After generating the dynamic sentence language model, all N-best sentence hypotheses from the first pass are rescored using the new model. The resulting sentence scores are combined with the scores from the first pass and the N-best hypotheses are reordered using the combination of scores. The weights for the scores can be optimized on a development set so as to minimize the word error rate.

This process must be executed for each utterance. In practice, the dynamic language model can be generated for a large batch of sentences, as long as the size of the dynamic vocabulary stays in the allowed bounds of the software and the size of the language model is reasonable.

The disadvantage of this method is that in the second pass, a new sentence-specific or sentence batch specific language model has to be built which makes it inappropriate for real-time applications like dictation. Another disadvantage is that a language-specific morphological analyser is needed for assigning a POS tag to each word, making it not directly portable to other languages.

Chapter 6

Evaluation

This chapter reports the results of experiments carried out in order to evaluate the usefulness of the proposed approaches. In the first section, speech, text and software resources for the experiments are described. The second section gives a detailed overview of a broad range of language modelling experiments. The third section reports results of some speech recognition experiments. In the final section, the experiments are summarized.

6.1 Resources for Estonian speech recognition experiments

Large speech and text resources are needed for training robust acoustic and language models for large vocabulary recognition. This section introduces the resources that were used in the evaluations.

6.1.1 Speech databases and their characteristics

This section describes the two speech databases used for speech recognition experiments. Database characteristics is given together with the explanation how the databases were partitioned into training and testing sets.

The BABEL phonetic database

The Estonian subset of the BABEL multi-language database [Eek and Meister, 1999] was collected in 1995-1998 at the Institute of Cybernetics at Tallinn University of Technology under EU COPERNICUS project ”BABEL - A Multilanguage Database”. The project was aimed at development of speech databases for six Central and Eastern European languages – Bulgarian, Estonian, Hungarian, Polish and Romanian. The database is based on the corpus design of EUROM_1 [Chan et al., 1995] with some modifications.

It was designed to fulfil different needs of phonetic research on Estonian sound system and prosody. The recordings were made in an anechoic chamber, directly digitized using 16-bits and a sampling rate of 20 000 kHz. The textual content of the database consists of numbers in the range of 0-9999, isolated CVC constructs, CVC-words in left and right context (i.e. word triplets), 5-sentence mini-passages and filler sentences. There are 55 different 5-sentence mini-passages, guaranteeing that all main phonologically relevant oppositions are revealed in the text corpus. The texts that were read are presented in Estonian orthography and in SAMPA phonemic transcription.

The main part of the database consists of three subcorpora. The ”many-talker set” has 60 speakers (30 male and 30 female). Each speaker read a set of isolated sentences, one or two mini-passages and 100 numbers. All speakers also read a set of CVC constructs. The ”few-talker set” consists of 8 speakers (4 male and 4 female). In this set, each speaker read ten mini-passages, a set of isolated sentences, 100 numbers. Two speakers also read three sets of CVC constructs. The ”very-few-talker set” has two speakers (a male and a female). Both of them read four sets of isolated sentences, 40 mini-passages and three sets of CVC constructs.

The main part of the database is distributed on three CD-ROMs and holds roughly 12 hours of audio data. 130 of the mini-passages (about 15% of all signals) have been manually segmented and labelled at phonemic level using SAMPA phonemic transcription.

In addition to the main part, there are so called ”set 2” and ”set 3” of the database that were recorded in 2001, with the original purpose of collecting data for speaker verification and identification research. Those sets feature 19 male and 15 female speakers who were each also present in the main part of the database. In both sets, each speaker reads one or two mini-passages, a set of isolated sentences and a set of numbers. For each speaker, the recordings for the two additional sets were done on the same day, with 15 minute intervals between the recording sessions. The additional sets are distributed on two CD-ROMs and contain around 7 hours of audio data.

Each of the mini-passages consists of five sentences with coherent semantic structure. The content of the sentences is mainly descriptive/conversational or simulates a situation of inquiry.

The filler sentences are designed by phoneticians to be especially rich in phonologically interesting variations. The sentences are also designed to reflect the syntactic and semantic complexity and variability of the language.

In recognition experiments, the many-talker set and the two additional sets are used for training acoustic models. The 138 isolated sentence utterances by six speakers in the few-talker set are used for evaluation. Two speakers in the many-talker set were not used since their utterances were designed to be used in word stress research and thus not really suitable for speech recognition experiments.

The SpeechDat-like speech database

The Estonian SpeechDat-like database was collected in 2002-2004 by the Institute of Cybernetics at Tallinn University of Technology [Meister et al., 2002, Meister et al., 2003, Meister et al., 2004]. The principles of corpus design, file formats, recording and labelling methods implemented by the SpeechDat [1] consortium were adopted and followed. The goal of the project was to collect speech from a large number of speakers for speech and speaker recognition purposes.

The database consists of speech recordings from voluntary speakers, recorded over the telephone line. The process of speaker recruitment and corpus collection was as follows: the project was advertised in different media channels where voluntary speakers were asked to register for a call at the homepage of the project. Each registrant then received a prompt sheet which contained instructions about the call, the list of the prompts, as well as the speech items to be read. During a call, each speaker was asked to speak the following 60 items:

- first name of the caller;
- spelled first name of the caller;
- birthday of the caller;
- birthplace of the caller;
- current week day;
- a spontaneous time phrase (answer to a question "What time is it?");
- four spontaneous answers to different "Yes/No" question
- a prompted 6-digit PIN-code;
- a prompted isolated digit string – 10 digits in random order;
- a prompted five digit number starting from 50000, which corresponded to the running number of the caller's prompt sheet;
- a prompted six digit number;
- a randomly generated phone number;
- a randomly generated credit card number with a valid checksum digit;
- a prompted time phrase;
- a prompted date phrase;
- a prompted relative and general time expression (e.g. *eelmisel reedel*, "last Friday");
- a prompted local city name;
- a prompted foreign city name;
- a prompted spelled word, words were drawn from a list of city names, person names and phonetically rich words;
- a prompted money amount;
- a prompted person name (surname and family name);
- eight prompted sentences;
- two prompted command-and-control application commands;
- two prompted phonetically rich words;

[1] `http://www.speechdat.org`

All calls were processed by manual auditive quality control. Only completed calls with adequate content were passed to the labelling stage. The labelling process was manual using semi-automatic processing software. Orthographic transcription was used.

The number of different speakers in the database is 1335. Around 100 speakers were asked to call at least 10 times using the same prompt sheet in order to collect data for speaker verification research. Thus, the number of calls is higher than the number of speakers, reaching 2969. The number of male and female speakers is almost equal. The age distribution of the speakers is roughly as follows:

- 13-22 years – 27% of speakers
- 23-32 years – 38% of speakers
- 33-42 years – 15% of speakers
- 43-52 years – 11% of speakers
- 53-62 years – 7% of speakers
- 63-72 years – 2% of speakers

Regional distribution shows that most of the speakers came from two largest cities in Estonia – Tallinn and Tartu. The rest of the speakers are quite equally distributed over other dialectal areas of Estonia.

About 41% of the calls came from fixed line network while the remaining 59% were made over cellular networks. A signal format of 8-bit A-law with a sampling rate of 8kHz was used for recordings.

For recognition experiments, the database was divided into training, development and test set. The development and test sets were chosen by randomly assigning 40 different speakers to each of the sets. To avoid using the same speaker's data for both training and evaluation, those 80 speakers were chosen out of those contributors who only made one call session. Only the prompted sentence utterances were used in evaluations, thus both the development and test set contained 320 utterances.

6.1.2 Text corpora and their characteristics

All text corpora used in this work has been compiled by the Working Group of Computational Linguistics at the University of Tartu. We use a the following subset of the Mixed Corpus of Estonian [Kaalep and Muischnek, 2005]:

- daily newspaper "Postimees", 33 million words,
- weekly newspaper "Eesti Ekspress", 7.5 million words
- Estonian original prose from 1995 onwards, 4.2 million words
- academic journal "Akadeemia", 7 million words
- transcripts of Estonian Parliament (Riigikogu), 13 million words
- weekly magazine "Kroonika", 600 000 words

The corpus contains further subcorpora (legislative documents, PhD dissertations) that are not used in this work because they were not regarded as suitable for

general large vocabulary language modelling.

Most of the texts has been downloaded from the web and processed and cleaned, i.e. only actual text content was stored and and navigational links and banners, etc. were removed. The general mark-up follows TEI Guidelines. The non-ASCII characters are represented as SGML entities. The texts are divided into paragraphs as in the original files. The text inside paragraphs has been processed so that the punctuation marks are separated from word forms by a space (except those punctuation marks that are an integral part of the token). Sentence boundaries have been automatically tagged with "<s>" and "</s>".

6.1.3 Software

For language model training and performance evaluation, SRILM toolkit [Stolcke, 2002] was used.

The HTK toolkit [Young et al., 2003] was used for automatic morpheme clustering for training class-based language models.

The SONIC Large Vocabulary Speech Recognition System 2.0-beta5 [Pellom, 2001] was used for training acoustic models and decoding test utterances. SONIC uses continuous density hidden Markov model (CDHMM) technology. The acoustic models are decision-tree state-clustered HMMs with associated gamma probability density functions to model state durations. Both manually created decision trees as well as automatically created trees are supported. The recogniser uses a two-pass search strategy. The first pass consists of a time-synchronous, beam-pruned Viterbi token-passing search through a lexical prefix tree. Cross-word acoustic models and trigram or four-gram language models are applied in the first pass of search. During the second pass, the resulting word-lattice is converted into a word-graph. Longer span language models can be used to rescore the word graph using an A* algorithm or to compute word-posterior probabilities to provide word level confidence scores. Sonic incorporates speaker adaptation and normalisation methods such as Maximum Likelihood Linear Regression (MLLR) [Legetter and Woodland, 1995], Parallel Model Combination (PMC), Jacobian Adaptation, and Vocal Tract Length Normalisation (VTLN) [Uebel and Woodland, 1999].

The Estonian morphological analyser [Kaalep, 1998b] and disambiguator [Kaalep, 1998a] by OÜ Filosoft was used for processing text corpora before language model training.

In previous work and experiments, the Julius decoder [Lee et al., 2001] and the CMU Sphinx 3 and 4 speech recognition engines[2] have been used.

For analysing and comparing speech recognition results, the NIST Speech Recognition Scoring Toolkit (SCTK)[3] version 2.1.4 was used.

[2] `http://cmusphinx.org`

[3] `http://www.nist.gov/speech/tools/index.htm`

In addition, many self-developed scripts mostly in the Perl programming language were used.

6.2 Language modelling experiments

This section reports results of a broad range of language modelling experiments. Such experiments aim to evaluate speech recognition language models using only linguistic analysis. The most common metric for language models is perplexity which measures the entropy of a statistical model against some reference data (see section 2.4.2). Another important measure is coverage of the language model vocabulary, usually measured in terms out-of-vocabulary words in some handout text.

First, vocabulary coverage is measured when using different kinds of basic units. The out-of-vocabulary rates of word, compound-split word, morpheme and minimum-length-constrained morpheme vocabularies is measured, with varying vocabulary sizes. With each experiment, various corpus statistics are investigated. Next, we analyse some heuristic techniques for improving the minimum-length-constrained morpheme vocabulary, as proposed in section 5.2. In the following section, several pseudo-morpheme language models are built, using varying span and different smoothing methods and parameters. Language model perplexities together with some other statistics are analysed. Then, we investigate the usefulness of interpolating language models built from different corpora. The next section describes the results obtained by clustering morphemes into classes based on corpus statistics. Finally, accuracy of the proposed compound word reconstruction technique is measured on both transcribed and recognised text, and its implications on the final word error rate are analysed.

6.2.1 Selection of basic units

In the following we evaluate the efficiency of language modelling using different methods for selecting basic units for language modelling. Main emphasis is put on analysing the vocabulary coverage with varying vocabulary sizes.

Words

The full training corpus contains 76 110 005 tokens, including sentence starting and ending tokens. The number of sentences is 5 569 936. The number of unique tokens is 2 126 765. After filtering out all words containing numerals and all words tagged as abbreviations or proper names, and after re-adding the top 500 most frequent words tagged as proper names, the number of different words is 1 652 961. Among those, 1 121 791 (68%) are compound words. Among all different compound words, 160 440 consist of more than two compound parts (14% of all different compound words). The most relevant (according to their

Table 6.1: Top 60 most common compound words in the training corpora together with their weights.

maa+ilma	479	esi+algu	144	ees+märk	101
vene+maa	393	vaba+riigi	138	oma+korda	100
võib+olla	380	iga+tahes	134	aja+lugu	99
pea+aegu	348	olu+kord	133	iga+üks	97
üli+kooli	321	kõige+pealt	131	vastu+pidi	96
nõu+kogu_de	257	maa+ilm	127	edas+pidi	94
see+tõttu	252	pea+minister	127	posti+mehe	94
maa+ilma_s	242	üle+jäänud	126	abi+kaasa	93
riigi+kogu	232	vahe+peal	124	linna+valitsuse	93
tähele+panu	211	mõni+kord	123	otse+kui	93
selle+pärast	210	see+kord	115	aeg+ajalt	92
esi+mees	206	posti+mees	115	lau+päeva_l	91
eel+kõige	184	linna+pea	114	olu+korda	90
tõe+poolest	177	saksa+maa	113	kas+või	89
see+järel	173	nii+siis	109	vana+ema	89
just+kui	168	voli+kogu	109	linna+valitsus	89
nõu+kogu	160	üks+kõik	107	oma+ette	88
posti+mehe_le	155	oma+vahel	107	nii+moodi	87
tõe+näolise_lt	153	vene+maa_l	107	aja+loo	86
see+pärast	144	nii+sama	101	nii+võrd	85

maximum likelihood score with regard to a test handout text) compound words in the training corpora are listed in table 6.1. The compound parts are separated using "+" and the optional morpheme ending is separated using the "_" character.

Words with compounds split

The large amount of compound words in the language creates an idea that we might split all compound words in the training corpus, and use the most common units as the vocabulary. Most compound word particles occur in the corpus also as independent words, so we might expect a nice overlap with words and compound word particles which should result in a shrinkage of the vocabulary size. Indeed, the number of different unique tokens after applying the vocabulary normalisation procedure is 645 059 – almost three times less than in the case of using pure words as basic vocabulary units. However, the OOV-rate of any reasonably sized vocabulary is still too high to be used as the basis of language modelling for large vocabulary speech recognition. Table 6.2 compares vocabulary coverages

for word and "*compounds-split*" based systems.

Table 6.2: Out-of-vocabulary rate of word-based vocabularies vs. split-compounds based word vocabulary.

Vocabulary size	Words	Words (compounds split)
10000	35.6	27.3
20000	28.6	20.0
30000	24.3	15.7
40000	21.7	13.0
50000	19.4	11.0
60000	17.6	9.5
100000	13.7	6.0
200000	9.1	3.0
400000	6.4	1.9

Morphemes

The morpheme vocabularies were constructed by preprocessing the training corpora using a script that separated all compound words and all suffixes from the preceding stems. All suffixes were tagged (using an underscore character) so that they would be treated separately from stems with the same orthography. After splitting, the number of tokens in the training corpora increased to 122 793 479. After applying the token set normalisation procedure as for other unit sets, the number of unique tokens is 154 605. The number of unique suffixes was 656. About one quarter of the suffixes seem to be mistakenly classified as suffixes but it shouldn't have a significant effect on the overall quality since such invalid suffixes occur very seldomly and won't have the chance to make it to the final recognition vocabulary: e.g., the number of suffixes in the maximum likelihood 40K vocabulary is 375. Table 6.3 lists the most relevant suffixes in the training data.

Morphemes with length constraint

As explained in chapter 5.1, one-phoneme morphemes are acoustically very confusable and thus can have a negative impact on recognition accuracy. The effect is amplified by the fact that most of such short morphemes occur very frequently and thus get a high unigram probability, which in turn increases the probability of the recogniser to insert them mistakenly in an incorrect place. Thus, we have opted to impose a constraint on the word splitting process that forces a morpheme to be at least two characters long, otherwise the splitting is not done.

Table 6.3: Top 45 most common morpheme suffixes in the training corpora together with their weights.

_s	26917	_id	5825	_gi	1817
_d	25978	_a	4901	_sin	1742
_b	16895	_vad	4861	_ti	1667
_t	15992	_mise	4201	_takse	1643
_l	14789	_tud	4022	_va	1575
_da	11843	_lt	3817	_na	1411
_ks	11198	_e	3356	_ja	1408
_st	10538	_sid	3269	_test	1362
_i	10215	_n	2568	_tele	1349
_nud	9999	_mis	2472	_tes	1306
_le	8448	_des	2337	_dagi	1263
_te	8211	_is	2208	_tel	1252
_ga	7029	_ta	2088	_me	1244
_ma	6195	_mine	1920	_lle	1127
_de	5946	_sse	1898	_dest	967

Of course, this increases the vocabulary size since all words that consist of a stem and a one-letter suffix must be treated as separate entries.

After processing the corpora, the number of tokens was 110923730. After normalising the vocabulary, the number of unique tokens was 223696. The number of unique tokens that were not split due to the length constrains was 82538. The number of unique suffixes grew from 656 to 918 – this is due to the fact that some of the suffix combinations were also treated as a single suffix due to length constraints (such as "_mis_t" in *tegemist*).

Analysis

The out-of-vocabulary rates for different basic units for language modelling with varying vocabulary sizes are shown on figure 6.1.

It is clear from the experiments that neither words nor compound-split words are suitable for language modelling using a conventionally sized vocabulary. The OOV-rate of the word-based vocabularies is much over what can be tolerated even when using a very large 800K size vocabulary. It can be seen that after splitting the compound words, the OOV-rate is roughly halved. Still, even when using a large 100K vocabulary, the OOV-rate is about 6% – too much to be used in large vocabulary speech recognition.

However, the OOV-rates of both morphemes and length-constrained

morphemes are much lower and can be compared with the OOV-rates of English word-based vocabularies of similar sizes. The OOV-rate of the morpheme-based vocabulary reaches the 2% threshold already when using a 40K vocabulary. While the OOV-rate of the morpheme-constrained vocabulary is 2-3 times higher than when using pure morphemes, it reaches a quite acceptable 2.40% when using a 60K vocabulary.

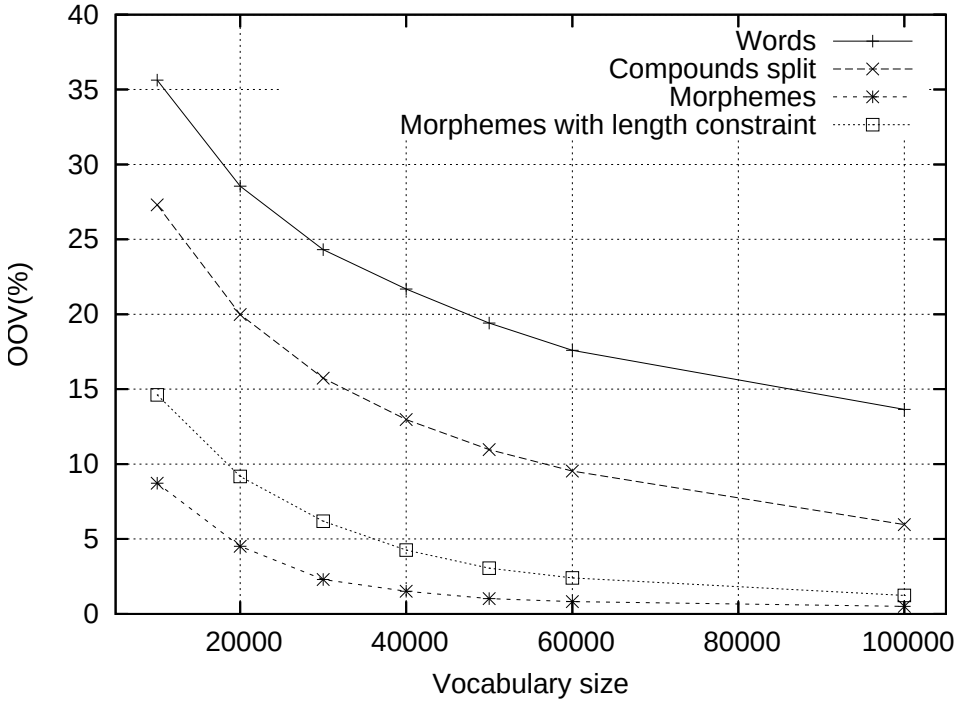

Figure 6.1: Out-of-vocabulary rates of different units for language modelling.

6.2.2 Vocabulary selection methods

In this section we evaluate methods that improve the language modelling vocabulary, as described in section 5.2. Out-of-vocabulary is measured against two test vocabularies: the sentence transcripts of the BABEL speech database test set (69 sentences), and the sentence transcripts of SpeechDat database that occur only in the test set (535 sentences). For weight tuning, the sentence transcripts of the SpeechDat database that do not occur in the test set were used (682 sentences).

A straightforward approach is to choose N most frequent words in the training data as the vocabulary. This method results in OOV rate of 3.59% for the BABEL test set and 3.06% for the SpeechDat test set.

Investigation of the resulting vocabulary shows that there are many words that contain non-native characters, such as numbers, punctuations and foreign letters.

Such words are not desirable from the speech recognition point of view since for most of such words we cannot generate the correct pronunciation. Thus, the first improvement step is to eliminate all such words from the vocabulary candidate list. This is done by generating a pronunciation for all words, and filtering out all such words whose pronunciation contains phonemes that are not in our phoneme set. This method decreases the overall OOV rate to 3.34% for the BABEL test set and 2.97% for the SpeechDat test set.

Further improvement is to filter out all words that the morphological analyser has tagged as abbreviations since the pronunciation generation script would attach a wrong pronunciation to them. This reduces the OOV rate to 3.25% for BABEL and to 2.91% for SpeechDat. A better approach would be to expand the abbreviations to the corresponding words before vocabulary selection but unfortunately the expanding is not trivial for languages like Estonian and it is not handled in this work.

Investigation shows that in the resulting vocabulary, there are a lot of proper names, especially among those units that occur less frequently in the corpus but still make it to the top 60 000 most frequent units. This is quite natural, as a most of the text corpus consists of newspaper texts, and newspapers tend to write a lot about certain persons, places and other subjects that have proper names. It is natural that in some, possibly rather limited period of time, one topic is discussed very frequently, and the talk about the topic contains many references to reoccurring names. However, we know that the speech recognition task that we have to deal with (recognising sentences from a speech corpus) does not contain a lot of proper names. Thus we tried the following *ad-hoc* approach to decrease the weight of proper names in the language model vocabulary: first, we filter out all words that are tagged as proper names by the morphological analyser; next, we re-add the most frequent 500 units that are tagged as proper names. Of course, this approach is quite specific to our task, and for other kind of tasks (e.g. broadcast news transcription), a very different method for handling proper names would probably be profitable. However, for our task, the described approach decreases the OOV-rate to 2.40% for the BABEL test set and to 2.07% for the SpeechDat test.

The final improvement is to apply the maximum likelihood based vocabulary selection technique as proposed in [Venkataraman and Wang, 2003]. For this, we need a sample text of the target domain to be used as heldout text, in addition to the test corpus used for OOV-measurements. We used the SpeechDat speech database sentence transcripts that were not used in the test set as such heldout text. This method decreases the OOV-rate to 2.05% for the BABEL test set but surprisingly increases the OOV-rate to 2.20% for the SpeechDat test set.

Table 6.4 summaries the OOV results after each optimisation phase.

Table 6.4: OOV for the 60 000 unit vocabulary after vocabulary selection improvement phases.

Vocabulary	BABEL OOV	SpeechDat OOV
Most frequent units from training corpora	3.59	3.06
Filter out un-pronouncable units	3.34	2.97
Filter out abbreviations	3.25	2.91
Allow only 500 most frequent proper names	2.40	2.07
Use heldout text for ML weight tuning	2.05	2.20

6.2.3 Morpheme-based N-gram language modelling

All language modelling experiments were performed with a minimum-length-constrained morpheme vocabulary that proved to provide the best speech recognition performance. The vocabulary size was 60 000 units and the units were selected from the union of the different text corpus vocabularies so as to maximise the likelihood of the development text (SpeechDat corpus training sentences that were not used for testing). Language modelling perplexity was measured against two test texts: BABEL speech database sentences and SpeechDat sentences that only occurred in the test set. The OOV-rate of the test texts is 2.05% and 2.20%, respectively.

The first language modelling experiments were made using the union of the text corpora as the basis for cut-off calculations and probability estimations. We built bigram, trigram and a few 4-gram language models, varying cut-off parameters from zero (i.e. singleton N-grams are included in the language model) to two (N-grams that occur two times or less are discarded). Table 6.5 shows the number of N-grams in the resulting language models and their corresponding hit rates (i.e their coverage) against the test texts. The N-gram hit rates show the percentage of N-gram requests that were found in the model and the percentage of N-gram requests that ended up in lower-order backed-off estimates. It was not possible to build a 4-gram language model that included singleton N-grams due to computer memory and processing time limitations.

The first observation from the results is that the hit rates differ dramatically between the two test sets – the N-grams probabilities needed by the BABEL test set can be much less frequently calculated from high order estimates than those of the SpeechDat test set. As much as 29.1% of the calculations on the BABEL set end up in unigram nodes when not including the singleton N-grams in the language model. The big difference is surprising, considering that the OOV-rate of the BABEL test set is actually lower than that of the SpeechDat test set. This phenomena could be explained by the fact that the sentences in the BABEL speech database are specially designed by phoneticians to be phonetically balanced. As a result they contain many rare words and many words in relatively uncommon

but perfectly grammatically legal order. The texts in the SpeechDat database are more similar to actual sentences in Estonian newspapers, magazines and books.

It turns out that the high order hit rates are also rather low for the relatively better performing SpeechDat test set. In comparison, the hit rates for an English 65K vocabulary 4-gram language model without singleton N-grams has been reported [Whittaker, 2000] to be 9.5/30.2/29.6/30.7, in comparison to 19.2/44.2/24.9/11.7 for the SpeechDat test set. The fact that 19.2% of all heldout N-gram probabilities are backed-off to the unigram estimates (when not using singleton N-grams) may be the cause of many recognition errors since the recognition errors are much more likely to occur within N-grams which have not been observed in the training data [Chase et al., 1994]. Such large difference between English and Estonian can be attributed to many factors, including the size of training data, heterogeneity of the test sets with regard to the training data and the relatively free order of the Estonian language.

Table 6.5: Language model size (number of bigrams/trigrams/4-grams) and hit rates of unigram/bigram/trigram/4-gram estimates using different language models with varying cutoffs.

Cutoffs	Number of N-grams	BABEL test set hit rates (%)	SpeechDat test set hit rates (%)
Trigram			
0/0	10M/37M	24.6/43.9/31.5	14.9/41.0/44.0
1/1	3.6M/7.0M	29.1/45.1/25.8	19.2/44.0/36.7
2/1	2.5M/7.0M	32.0/42.2/25.8	21.4/41.9/36.7
2/2	2.2M/3.6M	32.7/44.6/22.7	22.0/44.8/33.2
4-gram			
1/1/1	3.5M/6.5M/5.0M	29.1/45.4/18.5/6.9	19.2/44.2/24.9/11.7
2/2/2	2.1M/3.1M/2.1M	32.7/44.8/17.0/5.4	22.0/45.4/23.0/9.7

We also experimented with different discounting methods: Good-Turing discounting and Chen/Goodman's modified Kneser-Ney discounting [Chen and Goodman, 1998] as implemented in SRILM were applied for different language models. Katz back-off method [Katz, 1987] was used with Good-Turing discounting. The language model perplexity results are given in table 6.6.

The results show that the modified Kneser-Ney discounting algorithm gives clearly lower perplexity results than Good-Turing discounting. The difference was higher for the BABEL test set (19-26% relative) than for the SpeechDat test set (9-13%). The relative difference between the discounting methods was bigger when using trigrams than when using bigrams. As a result, only the modified Kneser-Ney discounting scheme will be used in all further experiments.

Another observation is that static interpolation of N-gram estimates with the

Table 6.6: Perplexities of different language models estimated over union of the training corpora.

Language model	BABEL test set	SpeechDat test set
Bigram, GT discounting, cutoff 0	1509	628
Bigram, GT discounting, cutoff 1	1566	667
Bigram, GT discounting, cutoff 2	1583	696
Bigram, mod-KN discounting, cutoff 0	1260	583
Bigram, mod-KN discounting, cutoff 1	1273	605
Trigram, GT discounting, cutoff 0/0	1317	490
Trigram, GT discounting, cutoff 1/1	1358	524
Trigram, GT discounting, cutoff 2/2	1377	555
Trigram, mod-KN discounting, cutoff 0/0	1008	438
Trigram, mod-KN discounting, cutoff 1/1	1003	453
Trigram, mod-KN discounting, cutoff 2/1	1011	470
Trigram, mod-KN discounting, cutoff 2/2	1017	479
Trigram, mod-KN discounting, cutoff 1/1, interpolated with lower order estimates	1050	463
Trigram, mod-KN discounting, cutoff 2/2, interpolated with lower order estimates	1068	492
4-gram, mod-KN discounting, cutoff 1/1/1	**988**	**436**
4-gram, mod-KN discounting, cutoff 2/2/2	1001	465

lower order estimates does not improve the perplexity results for either of the test sets. In fact, using interpolated estimates results in 2-5% higher perplexity than when using uninterpolated models.

The experiments show that retaining singleton N-grams in the language model is only sometimes useful in reducing the perplexity. For the bigram language model with modified Kneser-Ney discounting, retaining all N-grams lowered the perplexity for both the BABEL test set (1%) and the SpeechDat test set (4%). However, for the trigram language model, the presence of singleton N-grams improved only the perplexity of the SpeechDat test set (by 3%). The perplexity of the BABEL test set was by about a half percent higher when singletons were retained. These results correlate with earlier research for other languages [Whittaker, 2000] that suggests that singleton N-grams are generally useful in lowering the perplexity only if training corpora and test data are homogeneous. As was said in the analysis of the N-gram hit rates, the SpeechDat test data has more resemblance to actual sentences from newspaper and magazine articles and is thus more homogeneous to training data, while BABEL sentences contain many words in uncommon order and other peculiarities.

The 4-gram language models outperform trigram models with similar cutoff

parameters by around 1% for the BABEL test set and by 3-4% for the SpeechDat test set. In comparison, [Whittaker, 2000] reports a 8% improvement of 4-gram model perplexity over a trigram model for English and a 3% improvement for Russian, and speculates that the less significant perplexity improvement for Russian can partly be explained by the free word-ordering in Russian that makes the language model context increase less useful, since sequences of longer length are more likely to appear in the future as hitherto unobserved patterns. The same speculation can be made for Estonian. This is confirmed by the fact that the perplexity improvement for the BABEL test set (where many unusual word combinations occur) is lower than that of the SpeechDat set.

6.2.4 Interpolating domain-specific language models

In the previous section, language models were built using N-gram statistics from the union of the corpora. Instead of that, it is often profitable to build domain-specific language models from each domain-specific corpus and finally use linear interpolation to build a final model. This chapter shows some results of using this method.

Six different corpora as listed in chapter 6.1.2 were used. The interpolation coefficients for composing the final model were optimised on the SpeechDat database sentences that did not occur in the sentences used for testing. The optimisation process finds the weights that minimise the perplexity of the heldout text. The computation is iterative and stops when the interpolation weights change by less than a small threshold. The optimised weights depend on the language model cutoff parameters and the discounting methods. Optimised weights for two different language models are given in table 6.7. It can be seen that when singleton N-grams were included in the model (cutoffs 0/0), the weight of the smaller corpora turned out to be higher.

Table 6.7: Weights of different corpus-specific language models in the interpolated model.

Corpus	Optimised weights	
	Cutoffs 0/0	Cutoff 1/1
Postimees	58.7%	62.9%
Ekspress	15.9%	13.7%
Fiction	11.7%	13.1%
Akadeemia	9.1%	7.8%
Riigikogu	2.5%	1.4%
Kroonika	2.0%	1.1%

A set of different interpolated language models with various cutoff parameters was built. The perplexities obtained against the two test sets are listed in table 6.8.

Perplexities for equivalent language models built over union of the corpora can be found from table 6.6.

It turns out that for the BABEL test set, the use of this method always results in decreased perplexity. The improvement is especially significant (5%) for the language model that includes singleton N-grams which did not perform very well when estimated over the union of the corpora. The perplexities of the SpeechDat test set are actually higher when singletons are discarded. However, if singletons N-grams are retained, there is a 1% improvement in the perplexity also for this test set.

When using interpolation, including singleton N-grams in the language model is clearly profitable for reducing perplexity. It can be speculated that the interpolation provides an additional smoothing step that decreases the noise factor from including singleton events. Another reason might be the fact that not including singleton events in the case of interpolating domain-specific models also removes such events that occur twice or more in the union of the corpora but only once in each domain-specific corpora and has thus a harmful effect on the overall model accuracy.

The 4-gram model provided only marginally lower perplexity than the trigram model with the similar cutoffs for the SpeechDat test set, and didn't reduce perplexity for the BABEL set. This can explained by the fact that when including only those 4-grams that occur at least twice in one of the corpus, the number of 4-grams is not very high in general (3.6 million *vs.* 6.5 million when using union of the corpora). This reduces the already low 4-gram coverage even more and does not help much in making the language model more accurate.

Table 6.8: Perplexities of language models composed by interpolating domain-specific language models.

Language model	BABEL test set	SpeechDat test set
Trigram, mod-KN discounting, cutoff 0/0	**955**	**434**
Trigram, mod-KN discounting, cutoff 1/1	983	474
Trigram, mod-KN discounting, cutoff 1/1, interpolated with lower order estimates	999	470
4-gram, mod-KN discounting, cutoff 1/1/1	983	465

6.2.5 Class-based language modelling

In this chapter investigation on language modelling experiments is made using morpheme classes. Several class-based models are built using varying number of classes. The perplexity of the class-based models is measured as standalone as well as when interpolated with the morpheme-based model.

Experimental procedure

The clustering experiments were conducted using the word exchange algorithm implementation in the HTK toolkit. The vocabulary was the same 60 000 morphemes as was used in previous language modelling experiments. A list of unigram and bigram counts for all vocabulary morphemes was collected over the union of the training corpora. The initial classification placed the most frequent N_{C-1} vocabulary morphemes into their own class and the remaining vocabulary morphemes into the one remaining class. Sentence start and end symbols and all out-of-vocabulary morphemes were assigned to their own unique class and were not moved during the clustering procedure nor could other morphemes be placed into those classes.

Five clustering experiments were performed, with the number of classes N_C varying from 400 to 1200 with an increment of 200 to see the effect of the different number of classes on the class model. For each experiment, two clustering iterations were executed to decrease the danger of getting stuck in an unoptimal local minimum. After two iterations, all vocabulary morphemes had been moved up to two times between classes. The final classification function was then used to construct the class-based language model. The smoothed class trigram language model was constructed over the union of the corpora using modified Kneser-Ney smoothing as implemented in SRILM. Singleton class bigrams and trigrams were excluded from the model. The class membership component was estimated from training corpora using the empirical morpheme and the corresponding class unigram counts. No smoothing was applied.

Results

Perplexities of the five different class-based language models were computed against the BABEL and SpeechDat test sets, using the standalone class model and the interpolated class and morpheme trigram model. The morpheme trigram is our best-so-far model which is an optimised interpolation of corpus-specific models that use modified Kneser-Ney smoothing and include singleton N-grams. The perplexity of the morpheme-based model is 955 for the BABEL test set and 434 for the SpeechDat test set. The interpolation weights for the class-based and morpheme-based models were chosen so as to optimise the perplexity of the training set (the SpeechDat sentence transcripts that are not used in the test set). The EM algorithm was used for optimisation. The results are shown in table 6.9.

The interpolated class-based and morpheme-based model performed best when the number of classes was at least 800. It may be speculated that when using less classes, the class model is too general. The best results were obtained when the number of classes was 800 for the BABEL test set and 1200 for the SpeechDat test set, although there is not much difference when between performances with 800-1200 classes. The improvement over the pure morpheme-based model was

Table 6.9: Perplexities of standalone class trigram models and interpolated class and morpheme models.

Number of classes	Weights $\lambda_{morph}, \lambda_{class}$	BABEL		SpeechDat	
		PP_{class}	PP_{interp}	PP_{class}	PP_{interp}
–	1.00, 0.00	–	955	–	434
400	0.85, 0.15	1639	928	875	423
600	0.82, 0.18	1525	928	777	420
800	0.80, 0.20	1377	**906**	715	417
1000	0.77, 0.23	1326	914	680	415
1200	0.77, 0.23	1266	907	641	**414**

around 5% for both test sets. This is a less significant improvement than our previous results that reported a 7-8% improvement in perplexity [Alumäe, 2004a]. This can be explained by the fact that the earlier results used less language model training data (about 15 million words in comparison to 75 million words that we have now) and the class-based model is known to improve results most when training data is very limited.

It is interesting to look at contents of the statistically found morpheme clusters. Table 6.10 ten randomly chosen classes with all or up to ten of their most frequent members. Some classes (e.g. 659, 1198, 700) have members that are consistently semantically and functionally similar. In some classes (e.g. 1024, 1005), the top members have obvious semantic similarities with each other but the class also contains 'run-away' members that seem not to have anything to do with the more frequent members. Finally, the class 621 contains words that subjectively have no common attribute. Closer inspection reveals that the top word in the class, *ikka*, 'still', is about 1000 times more frequent than the second morpheme *ikki* (no meaning, possibly corrupted form of *ikka*, 'still'), and the rest of the three members occur 10 000 times less frequently and are probably assigned to this cluster by chance.

Table 6.11 compares language model unigram, bigram and trigram hit rates of the morpheme model (that includes singleton N-grams) and the class-based model. It is clear that the class model is indeed successful in drastically decreasing the number of calculations that end up in the unigram node, that was diagnosed as one of the reasons for the high perplexity in chapter 6.2.3.

6.2.6 Reconstructing compound words

This chapter describes the details of the implementation and experimental results using statistical compound word reconstruction.

Table 6.10: Ten randomly chosen morpheme clusters with all or up to ten most probable members from the 1200 class model.

Class 1024	Class 1005	Class 1003	Class 510	Class 621
küll	üle	kõrval	pika	ikka
kah	vajaka	sees	lühi	ikki
meelega	mehele	teel	pikema	sulad
agnostik	haute	juurest	lühikese	piuksus
igavles		kõrvalt	natukese	visar
servus		otsas	ktisise	
polüneeslane		_kesed	päristise	
		käigus		
		küljes		
		suus		
		tipus		
Class 1198	**Class 700**	**Class 659**	**Class 72**	**Class 792**
maksab	lille	miljonit	aja	juba
toetab	ranna	miljardit	tabloid	
tegeleb	kulla	triljonit	nägelikkuse	
ostab	liiva		luiske	
vastutab	muna		tributsiooni	
teenib	kalju		kobru	
suhtub	sini			
kaalub	linnu			
vastutav	soola			
kulutab	roosi			

Experimental procedure

For compound word reconstruction, a ”hidden event” language model was built. This language model is very similar to the trigram morpheme model used in perplexity experiments, with one additional vocabulary element that symbolises the hidden event of a compound border connector between two morphemes. The language model was estimated over the union of the corpora where the compound word connector tag was not filtered out.

The hidden event language model was applied to automatically insert the compound word connectors to a token stream that consists of morphemes. The result was compared with the reference data where compound word connectors between morphemes were retained. Insertion precision and recall were computed from the comparison. Precision is defined as a measure of the proportion of tags that the automatic procedure inserted correctly:

Table 6.11: Unigram/bigram/trigram hit rates of the morpheme-based model and the class-based model.

Model	BABEL	SpeechDat
Morpheme model	24.6/43.9/31.5	14.9/41.0/44.0
1200-class model	2.9/43.7/53.4	2.4/34.1/63.6

$$P = \frac{t_p}{t_p + f_p}$$

where t_p is the number of correctly inserted tags (true positives) and f_p the number of incorrectly inserted tags (false positives). Recall is defined as the proportion of actual compound word connector tags that the system found:

$$R = \frac{t_p}{t_p + f_n}$$

where f_n is the number of tags that the system failed to insert (false negatives).

Precision and recall can be combined into a single measure of overall performance by using the *F* measure which is defined as follows:

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}$$

where α is a factor which determines the relative importance of precision and recall. If we choose $\alpha = 0.5$, the *F* measure simplifies to

$$F = \frac{2PR}{P + R}$$

Another measure that is worth investigating is the word error rate that would result from the automatic compound reconstruction, given a perfect morpheme output by the decoder. The perfect output is of course impossible to achieve in reality since the OOV-rate of our vocabulary against the test texts is more than zero. The would-be word error rate is computed by applying the morpheme ending concatenation procedure, and the compound word recomposition procedure as described in chapter 5.4 and comparing the result with reference transcripts.

Results on reconstructing reference transcripts

As the first test, the method was tested on the reference transcripts from the BABEL and SpeechDat speech databases were used. The input consists of morphemes where compound word connectors are deleted.

The compound word connector tag insertion accuracy was measured when using bigram and trigram language models. Both language models were estimated over the union of the text corpora, using modified Kneser-Ney discounting.

Singleton N-grams were retained in the models. Tagging results are given in table 6.12.

Table 6.12: Compound word connector tagging accuracies and the resulting would-be word error rate resulting from incorrect tagging, given perfect morpheme output by the decoder.

Model	Test set	Inserted tags	Precision	Recall	F measure	WER
Bigram	BABEL	73	0.56	0.69	0.62	9.2
	SpeechDat	569	0.88	0.75	0.81	8.1
Trigram	BABEL	62	0.81	0.85	0.83	3.9
	SpeechDat	632	0.95	0.89	0.92	3.6

There is a very significant improvement when using a trigram model over the bigram model which is quite expected since the trigram model can capture the whole triple *word1 -compound- word2* which makes up the actual compound word. The tagging accuracy of the SpeechDat test set is much higher than that of the BABEL test set. However, the differences in the potential word error rate are not so big. This can be explained by the differences in compound word frequencies between the test sets: the BABEL test set contains 1307 morphemes boundaries and a compound connector should be inserted between 59 of them (about 4.5% of all cases); the SpeechDat test set has 7744 morpheme boundaries and 669 expected compound connectors (about 8.6%).

Table 6.13 lists some sentences from the SpeechDat test set that contain mistakenly compounded or uncompounded words. The errors are written in upper case and the correct words are written in the right column. Quick investigation reveals at least three common patterns where compound recomposition errors occur:

- a compound word is not recognised when both of the compound word particles are very infrequent: the result is that there is not enough occurrences of the pair, nor occurrences where the head word is a head in a compound, neither where the tail word is a tail in a compound; as a result, the statistical model has no reason to insert a compound connector between them (e.g. *piirde-tross*, *traks-tunkedes*, *ainu-autorsusest*, *broiler-küülik*)

- two words are sometimes mistakenly recognised as a compound word when the first word is often a head word in compound words, and/or the second word is often a tail word in compound words, although their pair may actually never occur as a compound, and it also doesn't occur as an uncompounded pair often enough (e.g. *suur laud / suur-laud*, *kuue meetri / kuue-meetri*)

- in some cases, words are mistakenly recomposed into a compound word when the fact that the words should be written separately comes from the surrounding context (e.g. *laulu looja / laulu-looja*, *kunsti tekke põhjuseks / tekke-põhjuseks*, *eri värvi osadest / värvi-osadest*). Those errors are probably the hardest to handle since the correct behaviour would often require understanding of the discourse.

Table 6.13: Some sample compound word connector tagging errors from the SpeechDat test set

Recognised	Actual
üles pannakse uued liiklusmärgid PIIRDE TROSS tõmmatakse pingule	.. PIIRDETROSS ..
ühevärviline kostüüm pikendab teie figuuri samas kui eri VÄRVIOSADEST lühendab	.. VÄRVI OSADEST ..
üheks kunsti TEKKEPÕHJUSEKS peetakse inimese tarvet ilu ja loomisrõõmu järele	.. TEKKE PÕHJUSEKS ..
väikeses ja pimedas kambris oli näha vaid voodi ja SUURLAUD	.. SUUR LAUD ..
väga soodsalt mõjuvad organismile tsitrused küüslauk ja TAIME SEEMNETES leiduvad ained	.. TAIMESEEMNETES ..
viis miljonit aastat tagasi VÄLJA SURNUD hiire fossiil oli üllatavalt hästi säilinud	.. VÄLJASURNUD ..
vaikne ja ennast ise kütusega varustav liikur on KUUEMEETRI pikkune silindriline puur	.. KUUE MEETRI ..
vaguniuksel istub taburetil õlistes TRAKS TUNKEDES naine	.. TRAKSTUNKEDES ..
vaesed MAA INIMESED said aru et see oli pogromm nende vastu	.. MAAINIMESED ..
LAULULOOJA oli huvitatud AINU AUTORSUSEST	LAULU LOOJA .. AINU-AUTORSUSEST
kuigi broileriks nimetatakse noort kana saab maitsva prae ka BROILER KÜÜLIKUST	.. BROILERKÜÜLIKUST

Results on reconstructing recognised words

The actual data that the compound word recomposition system has to work with is the recognised token stream from the decoder, not the perfectly tokenised reference transcripts. Therefore it is interesting to test the performance of the described method on actual recogniser output.

It is not obvious what to use as a reference measure for this test. The problem relates to the fact that the recomposition system cannot be expected to correctly recompose a compound word, if one of the word particles is not recognised correctly. For example, given a sentence

```
turbalademed ja kohalik maakki pälvisid uurijate
tähelepanu
```

and the recognised token stream

```
turba lademed ja kohalik maak _ki pälvi _sid uurijate
tähele PANNA
```

we cannot expect the tokens *tähele* and *panna* to be recomposed, since those two words are always written separately. As a solution, we regard a recomposition to be correct only if all of the compound word particles are recognised correctly, and the recomposition system composes them into a compound word. For this, an ”oracle” hidden event language model was trained on the reference transcripts of each of the test sets. When applied to the output of the recogniser, it effectively does what we need: inserts compound connector tags between correctly recognised compound particles.

Table 6.14 lists the compound recomposition accuracies when using the recognised token stream for the two test sets. The word error rate before compound recomposition (i.e. the compound particles were regarded as separate words) was 26.4 for the BABEL set and 36.0 for the SpeechDat set. After the automatic recomposition, the word error rate is 31.4 and 40.9, respectively. The word error rate after recomposing the compound words is almost always higher, since those two tests deal with two different token streams. If a pair of two correctly recognised compound word particles are recomposed correctly into one word, the correctly recognised word is counted as one, instead of two. Therefore, the weight of the correctly recognised words decreases. Of course, sometimes two incorrectly recognised words are recomposed into one word which decreases the word error rate.

Table 6.14: Compound word connector tagging accuracies and the resulting word error rate compared to the ”oracle” word error rate, given the actual recognised hypothesis from the decoder.

Test set	Precision	Recall	F measure	WER	Oracle WER
BABEL	0.60	0.85	0.70	31.4	28.0
SpeechDat	0.75	0.95	0.83	40.9	40.7

It is somewhat surprising that the final word error rate of the SpeechDat test set is only marginally higher than the corresponding oracle word error rate.

6.3 Recognition experiments

Speech recognition experiments using different language models and acoustic modelling techniques are presented in this section. Word error rate is used as the quality measure of the system.

First, an overview of the acoustic model training procedure is given. As the first recognition experiment, the performance of recognition with word, compound-split word, morpheme and minimum-length-constrained morpheme vocabularies is measured. Next, some different acoustic modelling techniques are investigated: we compare systems with grapheme-based and phoneme-based acoustic units; also, an experiment with an acoustic model that takes into account the effect of the overlong duration in Estonian language is conducted and the results are compared with a baseline system. Finally, we present results of an experiment that uses the two-pass method proposed in section 5.5.2.

6.3.1 Training and testing procedure

The SONIC toolkit [Pellom, 2001] was used in recognition experiments. SONIC uses decision tree state-clustered continuous density HMMs. The acoustic models have a fixed three state topology. Each state is modelled with a variable number of multivariate mixture Gaussian distributions. The training system uses the Viterbi algorithm for model estimation. Therefore, the training process consists of iteratively performing state-based alignment of the training audio, followed by an expectation-maximisation (EM) step in which HMMs are estimated. The frame-to-state alignments are considered fixed during each iteration of the EM algorithm. The initial alignment is performed using English acoustic models and a mapping from Estonian phonemes to English phonemes, as given in appendix A. In the next iterations, the training data is realigned using the acoustic models constructed in the previous iteration. We have found that five iterations of alignment and model estimation achieves adequate acoustic models. During the iteration of the EM algorithm, single-mixture triphones are estimated for each triphone occurrence in the training data. The estimated triphones are then placed at the root node of the decision tree of the corresponding phone and splitting questions are evaluated. The question that gives the largest increase in likelihood for the training data is applied for splitting the node. The splitting continues until the likelihood falls below a threshold or the number of frames assigned to a node becomes too small. Finally, the data assigned to each leaf node is used to estimate Gaussian mixture models. In our experiments, we configured the system to require at least 50 frames per mixture, use 2 to 24 mixtures per HMM state, and split the nodes if at least 300 frames per node remain.

During recognition, SONIC uses an algorithms based on the token passing model [Young et al., 1989]. Various methods to improve search efficiency are used [Pellom, 2001]. In our experiments, we configured the beam pruning so

that the recognition was executed in roughly 3 times slower than real time for the SpeechDat test set and almost exactly in real time for the BABEL test set. The experiments were executed on a machine with two dual-core Intel Xeon 3.2 MHz processors, however, during recognition, only single core was used by the decoder.

6.3.2 Comparison of different units for language modelling

We tested the impact of different basic language modelling unit selection methods to speech recognition word error rate. For this, the words in the training corpus were split in different ways and the 60 000 words with the highest maximum likelihood score were selected as the vocabulary. The corresponding language model was compiled by training six different corpus-specific trigram models (with singleton N-grams retained, using modified Kneser-Ney discounting) and interpolating the models using weights optimised on a heldout text (this method gives the best perplexity results as shown in chapter 6.2.4). The word-error rates were measured before reconstructing the compound words, i.e. compound word particles were regarded as different words in both the hypotheses and reference transcripts. In the experiment where full words where used, the recognised words were post-processed so that compound words were split. When using morpheme vocabularies, the endings were concatenated to the stems before scoring. The results are listed in table 6.15.

Table 6.15: Out-of-vocabulary rates and word error rates (without compound recomposition) when using different units for language modelling, with a vocabulary size of 60 000.

Units	OOV	BABEL WER	SpeechDat WER
Words (unsplit)	17.6	36.2	44.5
Split compounds	9.5	30.8	38.6
Morphemes	0.8	28.5	37.2
Length-constrained morphemes	2.4	26.4	36.0

It could be predicted that using shorter units as basic units gives better results than using whole words since the OOV-rate of the 60 000 full word vocabulary is almost 18%, compared to about 10% of the vocabulary where compound words are split. However, it is perhaps surprising that the vocabulary where only the compound words are split is only marginally worse than the vocabulary where words are split into morphemes, although the difference in OOV-rate is huge. The relative difference is about 8% for the BABEL and about 10% for the SpeechDat test set. Finally, the results confirm the claim presented in chapter 5.2 that omitting suffixes that consist of only one letter is advantageous: the language model of length-constrained morphemes results in a better WER than the model

of unconstrained morphemes for both tests sets, with relative differences of 9 and 3%.

6.3.3 Comparison of acoustic modelling techniques

In this section, we compare different acoustic modelling techniques. First, performance of grapheme-based acoustic models is compared with that of phoneme-based models. Then, simple phoneme-based models are compared with models where the vowel of the unstressed second syllable of an overlong foot is modelled using separate units. The performance is evaluated using WER before compound reconstruction, as in previous section.

Comparison of grapheme and phoneme based acoustic modelling

As discussed in section 3.2, Estonian orthography is almost phonetic with some notable exceptions. The exceptions occur mostly in the way plosives are written and pronounced. In section 4, a set of Estonian acoustic basic units for speech recognition was proposed. Also, an algorithm that can generate a pronunciation from word orthography and its morphological analysis was designed. To test the effectiveness and usefulness of this approach, we need to compare its effect on speech recognition word error rate with the baseline approach, where a word's pronunciation is directly produced from the orthography using a one-to-one mapping between word's graphemes and the phonemes in its pronunciation.

For this test, two different recognition systems were trained. The first one uses a set of acoustic models that directly corresponds to the Estonian alphabet, minus the letters *ž* and *z*. For all words, a simple grapheme to phoneme transformation was used, with a few exceptions: namely, the graphemes *ž*, *z*, *w*, *y*, *c* and *x* are transformed to phonemes /š/, /s/, /v/, /i/, /k/ and to a pair /k/ /s/, respectively. The words that included graphmes not available in the Estonian alphabet (plus the graphemes *w*, *y*, *c* and *x*) were not included as candidates for the language model vocabulary. If any of the acoustic model training utterances included such words, the corresponding utterances were removed from the training data.

For the second system, the grapheme-to-phoneme transformation script included the transformations as proposed in section 4, in addition to the few transformations that were also done for the first system. No discrimination for the palatalised and unpalatalised phonemes were made, i.e. they were regarded as one phoneme.

For both systems, a similar training procedure for both acoustic model and language model training was followed. The language model is length-constrained morpheme trigram language model that is an interpolation of domain specific trigrams as described in section 6.2.4. Singleton N-grams were not retained, modified Kneser-Ney smoothing was applied. Vocabulary was selected by taking the top 60 000 morphemes from the union of the domain-specific vocabularies

in order to maximise the likelihood of the SpeechDat sample sentences. The vocabulary of the two systems is not exactly the same because the second system uses pronunciation dependant morphemes and as a result, a morpheme that has one entry in the first system's vocabulary may have two entries in the second system's vocabulary. One such morpheme is the ending *_te* which is represented in the second system as two entries: *_te* [/t/ /e/] and *_te* [/tt/ /e/]. The amount of such double entries is very small but it may still have a certain impact on language modelling, since for the probability estimation of trigrams that include such morphemes, only the corresponding occurrences can be used.

The recognition word error results are listed in table 6.16. Clearly, there is not a large difference between the two systems. Moreover, for the BABEL test set the first system resulted in a better WER (3.5% relative difference) whereas for the SpeechDat test set, the second system gave slightly better results (1% relative difference).

Table 6.16: Word error rate for systems with orthography-based pronunciation dictionary *vs.* phonetic transcription based pronunciation dictionary.

Dictionary	BABEL	SpeechDat
Orthographic	27.3	37.2
Phonetic	28.2	36.8

Quantity degree specific acoustic models

As discussed in chapter 4.2, it might be useful to model vowels of the unstressed syllable of an overlong quantity degree foot as separate acoustic units.

To test this idea, all occurrences of vowels [a], [e], [i], [o] and [u] in the training transcripts were replaced with the corresponding extra short vowel, if they occurred in a second syllable of the overlong foot. This was rather straightforward to implement for the BABEL database, as the phonological transcripts already contained overlong syllable markers that were discarded in previous experiments. The correct shortened vowel was automatically found by searching for the first vowel after the overlong duration marker. If a compound word boundary or a word end was found before the vowel, the search was halted. For the SpeechDat database, a different approach was needed: all transcripts were reprocessed by the Estonian morphological analyser, that marked the places of overlong duration; then a similar shifting transformation as for BABEL transcripts was applied. Table 6.17 shows some words containing overlong foot together with their new pronunciations.

To test the new acoustic models using a large vocabulary recognition system, the pronunciations of all morphemes in the language model were to be checked. The goal was to detect all possible occurrences of overlong feet and use the extra-short vowel in the unstressed syllable, as in the training transcriptions.

Table 6.17: Pronunciation dictionary composition for some words containing an overlong duration.

Word	Morphological analysis with overlong duration markers	Pronunciation
enne, before	en:ne	$e\,n\,n\,e^{-}$
(ilusat) jaama, station, sg. gen.	jaa:ma	$j\,a\,a\,m\,a^{-}$
raudvarbade, iron pillar, pl. gen	rau:d+var:ba_de	$r\,a\,u\,t\,v\,a\,r\,p\,a^{-}\,t\,e$

This turns out to be a non-trivial task, as sometimes the morpheme boundary is located between the stressed and an unstressed syllable, as in word *'ükski'*, ':üks-ki' , 'none'. To solve this, all text corpora were processed as follows: each sentence was processed by the morphological analyser-disambiguator that marked occurrences of overlong feet; this information was used to produce the correct pronunciation for each morpheme of every word; for each resulting morpheme, it was checked whether the morpheme is in the language model vocabulary – if so, the given morpheme pronunciation was added to the pronunciation set of this morpheme. Finally, all discovered pronunciations were written to the dictionary. The resulting pronunciation dictionary of 60000 morphemes has 65348 different pronunciations in total. This means that 5348 morphemes (8.9% of all morphemes in the vocabulary) have two pronunciations. No morpheme had more than two pronunciations.

The results of recognition experiments are presented in table 6.18. As can be seen, the recognition quality of the BABEL-based system improved while that of the SpeechDat-based system degraded. The matched-pairs signed-ranks Wilcoxon test of statistical significance, as implemented in the NIST SCTK toolkit, shows for both systems that the difference in word error rate is insignificant.

Table 6.18: Word error rate when using baseline acoustic models *vs.* acoustic models with extra-short models for second syllable vowels in an overlong foot.

Acoustic models	BABEL	SpeechDat
Baseline	29.5	36.4
With extra-short models	27.4	37.6

Discussion

Both experiments with acoustic modelling and dictionary composition gave controversial results: the difference was not big and the system word error rate improved for one test set while degraded for the other test set.

It seems that the use of context sensitive hidden Markov models can fairly well model the variations in Estonian orthography and pronunciations and it is hard to significantly improve the system that uses simple grapheme-to-phoneme mapping for dictionary composition. It is not suprising that the grapheme-based models achieved a comparable performance with the phoneme-based models since similar results have been reported for other languages with a close grapheme to phoneme relation [Killer, 2003, Lihan et al., 2006].

Further significant improvements are probably possible and this should be one of the topics of future studies; in this work, further experiments with acoustic modelling were postponed as the process is especially time-consuming.

6.3.4 Language model improvements

Rescoring using morphological analysis

This section presents the results of rescoring N-best sentence hypothesis using morphological information and factored language model. The theoretical background of this approach was given in section 5.5.2. This work extends the experiments that were conducted earlier and reported in [Alumäe, 2006]. The performance is evaluated using WER of full words, as opposed to previous sections where compound particles were regarded as separate words.

The method was tested only on the SpeechDat data. This method needs a reliable and statistically relevant development set for tuning various parameters, thus the BABEL database with only six speakers in the test set was not suitable for this.

The baseline system uses our best performing length-constrained morpheme trigram language model that is an interpolation of domain specific trigrams as described in section 6.2.4. Singleton N-grams were retained, modified Kneser-Ney smoothing was applied. Vocabulary was selected by taking the top 60 000 morphemes from the union of the domain-specific vocabularies in order to maximise the likelihood of the SpeechDat sample sentences. Orthographic dictionary composition method was used.

Figure 6.2 presents the oracle WER values possible at various depths in a 1000-best list produced by the baseline system for the SpeechDat development and test set. The oracle uses reference transcripts to propose the best hypothesis at the given depth from the N-best list. Experimental analysis shows that the oracle can improve WER from 40.4 to 25.8 (36.1% relative improvement) for the development set and from 37.7 to 24.1 (also 36.1% relative improvement) for the test set. Most of the more accurate hypotheses are within the 100 N-best depth for each utterance: the relative oracle WER improvement at the depth of 100 is 30.9% and 31.8% for the two test sets, which is over 85% of the improvement from within 1000 hypotheses. The results suggest that we can gain substantial improvements by applying a strong post-processing and reranking mechanism to

the N-best lists, even at small N-best depth.

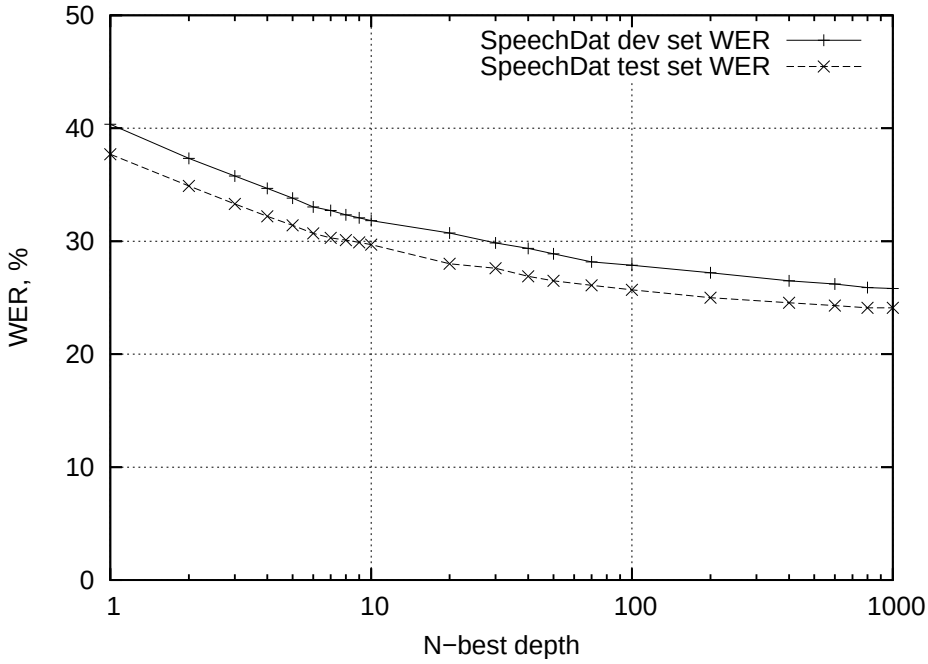

Figure 6.2: Oracle WER for the SpeechDat test set at various N-best depths.

The rescoring and reranking process consists of the following steps:

- All utterances in the test/development set are decoded using the baseline system. This results in a maximum of 1000 sentence hypotheses for each utterance (sometimes the pruning settings of the decoder limit the N-best list to a shorter length). For each hypothesis, an acoustic model and a language model score is recorded.

- All sentence hypotheses are processed using a hidden event language model that tags places of most probable places where compound words should be formed; the result is processed by a script that concatenates stems and suffixes and reconstructs compound words using the result from the statistical estimator. The baseline system stops at this point and outputs the hypothesis with the highest score.

- Now, all reconstructed hypotheses are processed by the Estonian morphological analyser and disambiguator that tags all words with their most probable part-of-speech (POS) tag.

- The resulting POS-tagged N-best lists are used to collect an N-best vocabulary of the current test set, that is, all words and POS-tags that occur in any

of the hypothesis;

- The resulting vocabulary is used to construct a factored language model, using an already morphologically tagged text corpus as training data. The factored language model is limited to the vocabulary that is actually needed for rescoring the current N-best lists and can thus be estimated and stored using a reasonable amount of memory and CPU time.

- After estimating the test-set specific factored language model, all N-best sentences for all utterances in the test set are rescored, that is, an additional language model score is added to each hypothesis.

- Next, the baseline acoustic model and language model scores and the additional factored language model score are combined using weights optimised on a development set. For finding the final output, a generalisation of the ROVER algorithm as implemented in SRILM is used that uses word error minimisation via dynamic programming and a voting process [Fiscus, 1997].

As mentioned, the scores combination weights are optimised on the development set. The optimisation process uses all available scores for each utterance, and the corresponding reference transcripts to find score combination weights so as to minimise the word error of a classifier that performs word-level posterior probability maximisation. A simplex-based "Amoeba" search [Press et al., 1988] on the (non-smoothed) word error function as implemented in SRILM is used. The search is restarted multiple times to avoid local minima.

The structure of the factored language model was tuned by hand in an *ad-hoc* manner so as to minimise the WER of the development set. The topology that achieves the best performance is outlined on figure 6.3. At first, the probability for each word in a sentence is attempted to be calculated based on the history of two previous words and their respective POS tags ($Pr(w_t|w_{t-1}, w_{t-2}, p_{t-1}, p_{t-2})$). As the POS tag is a deterministic function of the word in most cases (except for ambiguous words), this probability can be viewed as a standard trigram estimation ($Pr(w_t|w_{t-1}, w_{t-2})$). If the string $w_{t-2}\ w_{t-1}\ w_t$ did not occur in the training data, the model backs off by dropping the word w_{t-2} and tries to estimate the probability of a word, given the previous word and previous two POS tags. If such trigrams are still not found in the corpus, the model branches into 2 back-off paths by dropping the parent w_{t-1} or p_{t-2}, respectively, and using the mean score from the 2 branches as the final probability. Each branch tries to estimate its probability on the remaining factors, and backs off to use only one previous POS tag for calculating the probability. Finally, the model backs off to the unigram probability of the word.

The different conditional probabilities in each FLM node are examplified in table 6.19: given the trigram *koer tassib konti*, "dog carries a bone", the

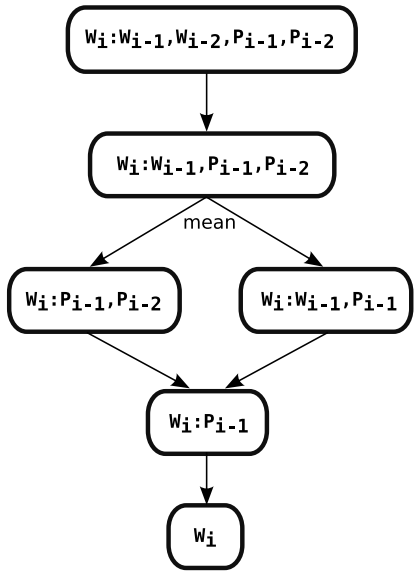

Figure 6.3: Backoff paths of the best-performing factored language model.

probability of the final word *konti* is first computed by looking for occurrences of *koer tassib konti*. If such trigram does not exist in the training corpus, we back off to the node below, and look for occurrences *(_S_ sg in) tassib konti*, i.e. by looking for trigrams where the word pair *tassib konti* is preceded by a noun in a singular nominative case. If such trigram still hasn't occurred in the corpus, we back off to two parallel back-off nodes, and calculate the mean from the result.

Table 6.19: Probability estimation in different nodes of the factored language model.

FLM node	Example of the probability estimation
$Pr(w_t \mid w_{t-1}, w_{t-2}, p_{t-1}, p_{t-2})$	$Pr(w_t = konti \mid w_{t-1} = tassib, w_{t-2} = koer, p_{t-1} = \_V\_ b, p_{t-2} = \_S\_ sgn)$
$Pr(w_t \mid w_{t-1}, p_{t-1}, p_{t-2})$	$Pr(w_t = konti \mid w_{t-1} = tassib, p_{t-1} = \_V\_ b, p_{t-2} = \_S\_ sgn)$
$Pr(w_t \mid p_{t-1}, p_{t-2})$	$Pr(w_t = konti \mid p_{t-1} = \_V\_ b, p_{t-2} = \_S\_ sgn)$
$Pr(w_t \mid w_{t-1}, p_{t-1})$	$Pr(w_t = konti \mid w_{t-1} = tassib, p_{t-1} = \_V\_ b)$
$Pr(w_t \mid p_{t-1})$	$Pr(w_t = konti \mid p_{t-1} = \_V\_ b)$
$Pr(w_t)$	$Pr(w_t = konti)$

The tests on rescoring the development set showed that it is not profitable to include all 1000 hypotheses for each utterance for reranking. Experiments showed

that the best WER reduction was achieved when around 100 best candidates were considered. At depths less and greater than that, the WER reduction began to degrade. Figure 6.4 presents a smoothed WER curves comparing the actually achieved results after rescoring with the oracle WER.

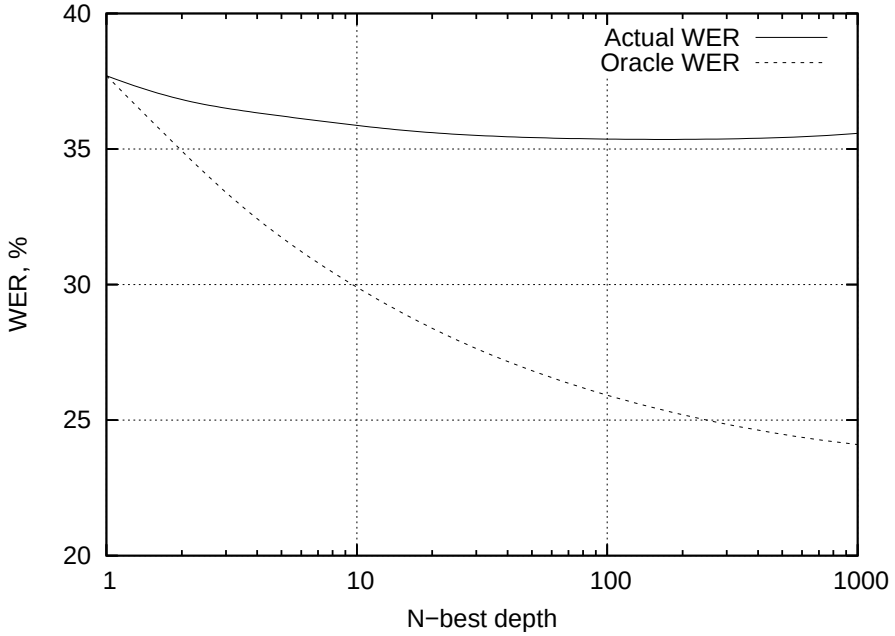

Figure 6.4: Actual WER of the development set after rescoring with the FLM compared to the oracle WER.

The WER results of the development and test set before and after rescoring are listed in table 6.20. To evaluate the effectiveness of the FLM approach, the rescoring experiments were also conducted using a conventional word trigram model, that is, a morpheme-based trigram model was used in the first pass, and the N-best lists were rescored using a word trigram model. This approach results in a small improvement for both sets (relative improvement of 4.5% for the development and 3.7% for the test set). Rescoring using the FLM improves the results further: relative improvement over the baseline results is 6.4% and 6.2%, respectively. As expected, the relative improvement is larger for the development set since the FLM topology and the score combination weights are optimised so as to minimise its word error rate and are not necessarily optimal for the test set.

In the results reported in earlier work [Alumäe, 2006], the relative improvement was larger (7.3% for the test set). However, the actually achieved absolute WER results were worse than reported here. This can be explained by the fact that the baseline language model used here is tuned better and the rescoring FLM can contribute less useful information for finding the best hypothesis.

Table 6.20: Word error rates for the basline system and after rescoring the 100 N-best hypotheses using a word-based trigram and a factored language model using POS-tags.

System	Dev set	Test set
Baseline	37.7	40.4
After rescoring with a word trigram	36.0	38.9
After rescoring with FLM	35.3	37.9

Table 6.21 lists hit counts for different back-off nodes in the rescoring factored LM described on Figure 6.3. The hit counts were computed by rescoring the 1-best sentences in the development set and looking were the probability calculation for each word ends up. Note that the lower nodes get an artificially higher hit count because the backoff path branches into two paths and always two nodes are hit if the second node is backed off from. The table also shows the amount of probability calculations that reached the corresponding nodes and could be actually computed without backing off to a lower level. Those percentages can be interpreted as being a usefulness metric of the nodes – the higher the percentage, the more calculations could be carried out in the corresponding node without using less accurate estimates of the lower level nodes.

Table 6.21: Hits counts for different backoff nodes in the rescoring FLM illustrated on Figure 6.3. The last two nodes can be reached from two branches, thus their hits are given separately.

Node	Hits	Percentage of "catches"
$Pr(w_t\|w_{t-1}, w_{t-2}, p_{t-1}, p_{t-2})$	754	24.1
$Pr(w_t\|w_{t-1}, p_{t-1}, p_{t-2})$	470	19.8
$Pr(w_t\|p_{t-1}, p_{t-2})$	824	43.3
$Pr(w_t\|w_{t-1}, p_{t-1})$	354	18.6
$Pr(w_t\|p_{t-1})$	560 + 1030	83.7
$Pr(w_t)$	521 + 521	100

There are less than 200 different part-of-speech tags in Estonian. While this number in much larger than the number of POS tags in English, it's still quite small. This created an idea that it might be advantageous to introduce an intermediate level of granularity between words and part-of-speech tags which would avoid a very steep drop in accuracy when backing off from words to POS tags. We selected 60 000 most frequent words from the union of the text corpora, and assigned them to 1000 clusters, using the word exchange algorithm as implemented in HTK. Bigram log-likelihood of the corpus was used as the optimization criteria. We then experimented with different FLM back-off topologies where words' classes would be used before backing off to the POS

tags. However, no improvement in WER was achieved.

Appendix B lists some sample recognised sentences together with the reference transcripts from the test set, after the rescoring technique has been applied.

6.4 Summary

In he following, a summary of the experimental work carried out throughout this chapter is given. We started with the simplest imaginable system, using grapheme-based acoustic models and a word-based trigram language model. The vocabulary was selected by taking the most frequent 60 000 units from the training corpus. The experimental steps together with the outcome are listed in table 6.22.

Table 6.22: Summary of system improvement steps.

Method	Result
Split compound words into particles	Improved OOV-rate (section 6.2.1) and WER (section 6.3.2)
Split compound word particles into morphemes	Improved OOV (section 6.2.1) and WER (section 6.3.2)
Don't separate one-grapheme morphemes from the preceding stem (*minimum-length-constrained morphemes*)	Worse OOV (section 6.2.1), improved WER (section 6.3.2)
Use heuristic methods in vocabulary selection in order to decrease the weight of proper nouns and eliminate abbreviations and words containing foreign characters	Improved OOV (section 6.2.2)
Select the vocabulary by weighting the different corpora using maximum likelihood count estimation, instead of taking the most frequent words from the union of the corpora	Improved OOV (section 6.2.2)
Create language model by interpolating corpus-specific models, include singleton N-grams	Improved perplexity (PPL) (section 6.2.4)
Interpolate N-grams with lower order estimates	No improvement in PPL (section 6.2.4)
Use 4-gram language model, discard singleton N-grams	No improvement in PPL (section 6.2.4)

Cluster morphemes using text corpus statistics, interpolate class-based languge model with morpheme-based language model	Improved PPL (section 6.2.5)
Use hidden event language model for compound word reconstruction	Able to reconstruct most compound words (section 6.2.6)
Use phoneme-based acoustic models	No consistent improvement in WER (section 6.3.3)
Use phoneme-based acoustic models, create separate models for short vowels in unstressed syllables of ovelong foot	No consistent improvement in WER (section 6.3.3)
Use dynamically built word-based trigram for rescoring 1st pass hypotheses	Improved WER (section 6.3.4)
Use POS-tags in the rescoring process for backing-off	Improved WER (section 6.3.4)
Use statistically derived word classes in the rescoring process for backing-off	No improvement in WER (section 6.3.4)

Chapter 7

Conclusion

This thesis presents an investigation into large vocabulary continuous speech recognition modelling (LVCSR) issues for Estonian. In particular, it describes approaches to acoustic and language modelling in the context of a contemporary statistical framework. The application of models in a standard hidden Markov model (HMM) based speech recognition system should make porting an already available recognition system to Estonian easier and cheaper. Basic concepts of modern speech recognition were reviewed in chapter 2. Overview of Estonian phonology, orthography, morphology and syntax was given in chapter 3. An introduction to the three-way duration system in Estonian phonology was also presented.

7.1 Review of the study

In chapter 4, an approach to Estonian acoustic modelling for LVCSR using HMMs was presented. First, the inventory of acoustic units was proposed. The units correspond roughly to Estonian short phonemes, as described in chapter 3. Long phonemes that occur in long and overlong feet are modelled by sequences of two corresponding phone units. This was justified by the common judgement that there is little difference in the quality of short and long Estonian phonemes. The abundance of diphthongs in Estonian makes it unfeasible to model each diphthong by a separate unit, as opposed to languages like English. Therefore, the modelling of diphthongs using sequences of two basic phone units should be beneficial. This also complies with the handling of long vowels whose usage is similar to diphthongs in Estonian. The only exception in the handling of short and long phonemes lies in the modelling of plosives since the realization of long plosives is clearly different from concatenation of two short plosives. Therefore, we proposed to model short and long plosives (including realizations in an overlong foot) using separate units. Pairs of palatalised and unpalatalised phonemes are merged into one acoustic unit since there is no difference in orthography of such

oppositions and the determining of the palatalisation from the written form of a word is unreliable. The chapter also proposed an alternative set of acoustic units where the short quality-degraded vowels of the unstressed second syllable in an overlong foot are modelled using separate units. Finally, a simple grapheme-to-phone transformation algorithm was developed. The algorithm relies on tagged compound particle borders since compound borders have an important effect on the way plosives are pronounced.

In chapter 5, a technique for large vocabulary language modelling for Estonian was developed. The approach relies on automatic morphological analysis and the treatment of morphemes as basic language units. The morphological analyser is used in processing training corpus to split compound words into particles and separate morpheme endings from stems. A maximum likelihood vocabulary of the resulting particles is used as the language model lexicon. To reduce acoustic confusability, we proposed to split morphemes only if the resulting length of the particles is at least two graphemes. Some heuristic techniques for improving the vocabulary selection were presented. After decoding with the morpheme-based N-gram model, the suffixes in the decoder output hypotheses are reattached to the preceding stems. A separate hidden event language model is applied for reconstructing compound words, using only lexical correlates of compound word particle boundaries. Finally, two independent methods for improving the language model were introduced. The first method uses corpus statistics to assign morphemes into clusters and thereby make morpheme N-gram probability estimates more robust. The second method uses a two-pass strategy to apply a word-based N-gram model in the second pass. The word-based model uses morphological part-of-speech tags in a backing off scheme to improve robustness and reduce local morphosyntactic errors in recogniser output.

Various language modelling and speech recognition experiments were presented in chapter 6. First, the coverage of different basic units for language modelling units was measured. The experiments confirmed that both words and compound-split words are not suitable units for language modelling when using a conventionally sized vocabulary of 60 000 units – they produced out-of-vocabulary (OOV) rates of about 18% and 9%, respectively. However, pure morphemes and minimum-length-constrained morphemes achieved an acceptable OOV rate of 0.8 and 2.4%, respectively. The next experiments focused on building N-gram models using minimum-length-constrained morphemes as basic units. The best perplexity results were achieved when a trigram was composed by interpolating corpus-specific models. In this case, including singleton N-grams in the model was clearly beneficial. Perplexities of the best-performing models against two heldout texts were 955 and 434, respectively. The perplexity could be further improved to 906 and 414, respectively, by interpolating the morpheme-based model with a class-based model. The last set of language modelling experiments was carried out to investigate the proposed compound word recognition method. First, the compound word connector

tagging accuracy was measured on transcribed speech. The F-measure describing the precision and recall of the process was 0.83 and 0.92 for two test sets. When using recognised text, the corresponding figures were lower as expected – 0.70 and 0.83. Some interesting error patterns where compound recomposition errors occur were identified. The second part of the chapter describes various speech recognition experiments. First, word error rates (WER) were measured when using different units for language modelling. The experiments confirmed earlier claims that it is profitable not to introduce suffixes that consist of only one grapheme – the model using minimum-length-constrained morphemes achieved the best WER. Next, some experiments with different acoustic models were conducted: a system using phoneme-based acoustic models was compared with one using grapheme-based models; also, a system using separate models for short quality-degraded vowels of the unstressed second syllable in an overlong foot was investigated. Those experiments gave controversial results and the difference between the systems was not significant. We concluded that it would be difficult to improve simple grapheme-based acoustic models since Estonian is a language with a close grapheme to phoneme relation. The last set of recognition experiments investigated the proposed two-pass recognition strategy that applies a dynamically-built word-based factored language model using morphological information for rescoring N-best sentence hypotheses from the first pass. A relative WER improvement of 6.2% was achieved. The final WER of the development and test set was 35.3% and 37.9%, respectively.

7.2 Future work

There are several aspects of the work presented in this thesis that will require further investigation.

In the domain of acoustic modelling, the biggest challenge lies in proper modelling of the tree-way quantity opposition in the accented positions within the rhythmic foot of Estonian words. Listening tests have shown that the three quantity degrees cannot be identified on segmental or syllabic levels. Rather, the most important feature in identifying the quantity degree is the duration ratio of the first stressed and the second unstressed syllable. Of course, such feature cannot be adequately modelled using phoneme-level hidden Markov models. The easiest way to identify the correct quantity degree during the recognition process would probably be to use conventional acoustic models in the first pass, align recognised hypotheses with the audio and calculate the duration ratios directly from there. More interesting and potentially more beneficial would be integrating the relative duration analysis directly into the first pass.

In the domain of language modelling, many interesting research directions could be investigated. First, the preprocessing and normalisation methods of training text corpora can be much improved. Currently, words containing numbers as

well as abbreviations are completely ignored. Instead, numbers and abbreviations should be expanded into words. This is however not trivial since the expansions should take into account the inflection of the numbers and abbreviations in the given context.

The splitting of words into morphemes using a morphological analyser should be compared with some data-driven methods, e.g. using the minimum description length principle as has been done for Finnish and Turkish [Siivola et al., 2003, Creutz and Lagus, 2005]. First experiments with Estonian recognition have already been reported [Hirsimäki et al., 2006] but they didn't compare the data-driven method with the morphological analysis based system.

The proposed compound word reconstruction technique could be improved. The analysis of reconstruction errors revealed two kinds of problems caused by data sparseness issues. Some of such issues could probably be eliminated by using a class-based language model. An added area for further study is to combine acoustic and prosodic cues, such as pause length, phone duration and pitch around the boundary between possible compound particles, with the linguistic model, as has been done for automatic sentence segmentation [Stolcke et al., 1998, Shriberg et al., 2000].

In section 6.2.3 we discussed that the high perplexity of the morpheme-based language model can be at least partly blamed on the low coverage of the trigram model and the relatively high amount of trigrams in heldout texts that end up in being estimated in unigram nodes. This is caused by the limited amount of language model training data for Estonian and the relatively free word order of the language. This suggests that in order to make progress in recognition results, we need to find ways to improve the coverage, span and robustness of the language model. In the language modelling community, there have been recently great interest in efforts that study various ways of using information from a longer context span than that usually captured by normal N-gram language models, as well as ways of using syntactical information that is not available to the word-based N-gram models [Chelba and Jelinek, 2000, Charniak, 2001]. Another recent work that has shown promising results uses distributional representation of words and neural networks for language modelling [Bengio et al., 2000, Schwenk and Gauvain, 2005]. This method has the ability to accommodate longer contexts and has been shown to significantly improve on regular N-gram models in perplexity. Such techniques could be also applied for morpheme-based language modelling.

The two-pass recognition method proposed in this thesis uses a word and part-of-speech based factored language model (FLM) to rescore the hypotheses from the first pass. Instead of an FLM, or in addition to it, other kinds of sentence probability estimators could be used in the second pass. One approach worth investigating is the use of latent semantic analysis (LSA) based language model [Bellegarda, 2004]. The use of word particles as modelling units makes it very difficult to integrate LSA-based language model into the first pass decoder. Also,

huge number of different words in the language make the standard LSA approach probably quite ineffective. However, in the second pass, we could use word stems as LSA modelling units. The number of different stems is much less than the number of different inflected word forms, and given the long-distance nature of the LSA technique, they are more suitable modelling units than the actual words.

Finally, it would be highly interesting to apply the proposed methods to a more practical speech recognition applications. During the described experiments, only the read sentences from two speech corpora were used for decoding. The set of sentences was designed by phoneticians so as to achieve a high coverage of all phonemes in different contexts, and therefore contain a lot of rare words in rare contexts. As a result, the sentences, especially those in the BABEL corpus, have an extremely high perplexity (over 900). This suggests that much lower language model perplexities could be achieved for real world sentences. On the other hand, the sentences in the speech corpora do not contain many proper nouns, numbers and dates, as opposed to typical sentences in texts like broadcast news. Therefore, it can be predicted that new kinds of challenges (e.g. adding new proper nouns to vocabulary, handling of foreign names) would turn up when porting the recognition system to practical tasks.

7.3 Summary

This chapter has summarised the conclusions from this study and identified areas for future research. The thesis proposed methods for Estonian large vocabulary continuous speech recognition and evaluated them with various language modelling and speech recognition experiments. An inventory of phoneme-based acoustic models was proposed. Experiments showed that simple grapheme-based context-sensitive acoustic models achieve a comparable performance with the phoneme-based models. The treatment of minimum-length-constrained morphemes as basic units for language modelling was suggested, its benefits were confirmed with evaluations. A proposed statistical model for compound word reconstruction was shown to achieve high accuracy. A two-pass approach using a dynamically-built word-based language model for rescoring first pass hypotheses demonstrated promising results. Future work should aim at improving the coverage, span and robustness of the language model using methods beyond N-gram model. Also, porting the recognition system to more practical tasks could reveal new challenges.

Abstract

The goal of large vocabulary automatic speech recognition is to recognize natural language as spoken by humans, and convert it to textual representation. There are many application areas for such technology, such as desktop dictation, automatic transcription of radio broadcasts and audio archives and automatic close-capturing of television broadcasts. The use of statistical and data-driven methods has resulted in great progress in this technology. For many languages with large number of speakers, many successful recognition systems have been developed. Due to limited resources and technological complexity, Estonian large vocabulary speech recognition hasn't been available. However, it is highly important for the survival of small languages to develop human language technologies for the language, including tools for both text and speech processing.

This dissertation concerns the development of methods and models for use in large vocabulary automatic speech recognition system for Estonian. The presented approach adapts the modern general-purpose statistical framework using hidden Markov models for modelling variation of basic speech sounds and statistical N-gram models for approximating natural language. The thesis concentrates on the language specific design issues of the three knowledge sources that are applied during recognition: the acoustic model, the language model and the pronunciation lexicon.

The proposed set of acoustic units corresponds roughly to Estonian short phonemes. Long phonemes as well as diphthongs, geminates and consonant clusters are represented by sequences of two or more corresponding short units. Short and long plosives are modelled using separate units. Pairs of palatalised and unpalatalised phonemes are merged into one unit. No distinction between long and overlong duration is made. A grapheme-to-phone transformation algorithm is presented.

Estonian is a highly inflective and compounding language with practically unlimited number of different word forms. The large number of unique words makes the conventional statistical language modelling approach not suitable for Estonian: given a set of 60 000 most frequent word forms, it is difficult to achieve satisfying word coverage, and it is hard to robustly estimate word probabilities in different contexts from sparse training data. The technique presented here relies on automatic morphological analysis and the treatment of morphemes as

basic language units during the recognition process. After decoding, recognized morpheme suffixes are concatenated back to the preceding stems. A separate statistical model using only lexical information is applied for compound word reconstruction. In addition, two independent methods for improving the language model are proposed. The first method clusters morphemes into classes based on text corpus statistics, in order to make N-gram estimates more robust. The second method uses a two-pass strategy to apply a dynamically built word-based language model in the second pass. Statistical morphological correlates are used for reducing local morphosyntactatic errors.

The proposed techniques are evaluated on a range of language modelling and speech recognition experiments. The proposed phoneme-based acoustic models are found to perform similarly with the simpler grapheme-based models. Language modelling experiments indicate that using subword units is beneficial for Estonian large vocabulary tasks. This result is confirmed with recognition experiments. The proposed two-pass method further improve the results.

Keywords: Estonian, speech recognition, LVCSR, agglutinative languages, highly inflective languages, morphemes, compound words, class-based models, factored language models

Kokkuvõte

Suure sõnavaraga automaatse kõnetuvastuse eesmärgiks on loomuliku inimkõne tuvastamine ja selle teisendamine tekstiks. Sellisel tehnoloogial on palju rakendusalasid, näiteks dikteeritud teksti viimine tekstikujule, arhiveeritud raadio- ja muude kõnearhiivide automaatne transkribeerimine, automaatne subtiitrite genereerimine telesaadetele. Tänu statistilistele ja andmepõhistele meetoditele on selles valdkonnas tehtud suuri edusamme. Mitme suurema keele jaoks on olemas edukalt töötavaid suure sõnavaraga tuvastussüsteeme. Ressursside vähesusest ja tehnilisest keerukusest tingituna pole eesti keele puhul selliste rakendusteni seni veel jõutud. Kõne- ja tekstitöötlusvahendite arendamine on väikese keele ellujäämist silmas pidades aga väga oluline.

Käesoleva väitekirja peamiseks uurimisobjektiks on suure eestikeelse sõnavaraga kõnetuvastus. Töös käsitletakse eesti keelele sobivaid mudeleid ja meetodeid, kasutades kaasaegset üldkasutatavat statistilist lähenemist, mis modelleerib kõneühikute varieeruvust Markovi peitmudelitega ja loomulikku keelt statitilise N-gram mudeliga. Töö keskendub kolme keele-spetsiifilise tuvastuses kasutatava mudeli – akustilise mudeli, keelemudeli ja häälduss̃õnastiku – tehnilise lahenduse väljatöötamisele.

Töös välja pakutud akustiliste ühikute hulk vastab umbkaudu eesti lühikestele häälikutele. Pikki vokaale, diftonge, geminaate ja konsonandiklastreid modelleeritakse mitme lühikesele foneemile vastava ühikuga. Lühikesi ja pikki sulghäälikuid modelleeritakse eraldi ühikutega. Palataliseeritud ja mittepalataliseeritud häälikud vastavad samale ühikule. Teises ja kolmandas vältes esinevat pikka häälikut modelleritakse sarnaselt. Sõnade teisendamiseks ortograafiliselt kujult akustilistele ühikutele vastavale kujule saab kasutada välja pakutud teisenduslgoritmi.

Eesti keele morfoloogia on suurel määral flektiivne-aglutinatiivne, samuti kasutakse palju liitsõnu. Seetõttu on erinevate keeles esinevate sõnavormide arv praktiliselt lõpmatu. Tänu suurele unikaalsete sõnavormide hulgale ei tööta standardne statistiline keelemudel eesti keele puhul kuigi hästi: kasutades ainult 60 000 keeles kõige sagedamini esinevat sõnavormi, ei ole võimalik saavutada head keelemudeli katvust. Lisaks sellele, erinevate sõnakombinatsioonide usaldatava esinemistõenäosuse leidmine ei ole sõnade rohkuse tõttu võimalik. Selles töös arendatud meetod tugineb automaatsele

morfoloogilisele analüüsile ning käsitleb tuvastuse käigus morfeeme keele põhiühikutena. Pärast tuvastust liidetakse lõpud eelnevatele tüvedele. Liitsõnade rekonstrueerimiseks rakendatakse eraldi statistilist mudelit, mis tugineb ainult leksikaalsele informatsioonile. Töös pakutakse välja kaks eraldiseisvat meetodit keelemudeli kvaliteedi parandamiseks. Neist esimene klassifitseerib kõik morfeemid klastritesse, kasutades tekstikorpuse statistikat. Klasterdamine suurendab keelemudeli usaldsuväärsust. Teine meetod koosneb kahest faasist: esimeses kasutatakse kirjeldatud morfeemidel põhinevat keelemudelit; teises faasis konstrueeritakse dünaamiliselt sõnadel põhinev keelemudel ja antakse selle abil esimesest faasist saadud lausekandidaatidele uued hinnangud. Lokaalsete morfosüntaktiliste vigade vähendamiseks kasutatakse sõnade ja sõnaliikide statistikat.

Kirjeldatud mudeleid ja meetodeid testitakse erinevate keelemodeleerimise ja kõnetuvastuse eksperimentidega. Osutub, et hääliku-sarnaseid akustilisi ühikuid kasutades saavutatakse tähe-sarnaste ühikute kasutamisega sarnane tuvastustäpsus. Keelemodelleerimiseksperimendid näitavad, et morfeemide kasutamine on eesti keele suure sõnavaraga tuvastussüsteemis kasulik. Seda kinnitavad ka kõnetuvastuseksperimendid. Kasutades väljapakutud kahefaasilist meetodit, on tuvastustäpsust võimalik veelgi parandada.

Võtmesõnad: Eesti keel, kõnetuvastus, flektiivne keel, aglutinatiivne keel, morfeemid, liitsõnad, klassi-põhised mudelid, faktoritel põhinev keelemudel

Bibliography

[Alumäe, 2002] Alumäe, T. (2002). Piiratud sõnavaraga eesti keele kõnetuvastus. Master's thesis, Tallinn Technical University, Tallinn, Estonia.

[Alumäe, 2003] Alumäe, T. (2003). Eestikeelse kõne tuvastus: prototüübi loomine. In *Eesti Keele Instituudi toimetised 12. Toimiv keel I*, pages 34–49, Tallinn. Eesti Keele Sihtasutus.

[Alumäe, 2004a] Alumäe, T. (2004a). Large vocabulary continuous speech recognition for Estonian using morpheme classes. In *Proceedings of ICSLP 2004 - Interspeech*, pages 389–392, Jeju, Korea.

[Alumäe, 2004b] Alumäe, T. (2004b). Large Vocabulary Continuous Speech Recognition for Estonian Using Morphemes and Classes. In *Proceeding of the 7th International Conference, TSD 2004*, pages 245–252, Brno, Czhech Republic.

[Alumäe, 2005a] Alumäe, T. (2005a). Phonological and morphological modeling in large vocabulary continuous Estonian speech recognition system. In Langements, M. and Penjam, P., editors, *The Second Baltic Conference on Human Language Technologies : Proceedings*, pages 89–94, Tallinn, Estonia. Institute of Cybernetics (Tallinn University of Technology); Institute of the Estonian Language.

[Alumäe, 2005b] Alumäe, T. (2005b). Using Adaptive Stochastic Morphosyntactic Language Model for Two-pass Large Vocabulary Estonian Speech Recognition. In *Proceedings of SPECOM 2005*, pages 515–518, Patras, Greece.

[Alumäe, 2006] Alumäe, T. (2006). Sentence-adapted factored language model for transcribing Estonian speech. In *Proceedings of ICASSP*, volume 1, pages 429–432, Toulouse, France.

[Alumäe and Võhandu, 2003] Alumäe, T. and Võhandu, L. (2003). Piiratud ulatusega eestikeelne kõnetuvastus. In *Eesti Keele Instituudi toimetised 12. Toimiv keel I*, pages 50–52, Tallinn. Eesti Keele Sihtasutus.

[Alumäe and Võhandu, 2004] Alumäe, T. and Võhandu, L. (2004). Limited-vocabulary Estonian continuous speech recognition system using hidden Markov models. *Informatica*, (3):303–314.

[Ariste, 1939] Ariste, P. (1939). A quantitative language. *Proceedings of the 3rd International Congress of the Phonetic Sciences*, pages 276–280.

[Ariste, 1953] Ariste, P. (1953). Foneem eesti keeles. *Eesti NSV Teaduste Akadeemia Toimetised, Ühiskonnateaduste Seeria*, (3):357–367.

[Bellegarda, 2004] Bellegarda, J. R. (2004). Latent semantic language modeling for speech recognition. In Johnson, M., Khudanpur, S., Ostendorf, M., and Rosenfeld, R., editors, *Mathematical Foundations of Speech and Language Processing*, pages 73–104. Springer Verlag, New York, USA.

[Bellegarda, 2005] Bellegarda, J. R. (2005). Unsupervised, language-independent grapheme-to-phoneme conversion by latent analogy. *Speech Communication*, 46:140–152.

[Bengio et al., 2000] Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. In *Advances in Neural Information Processing Systems*, pages 932–938.

[Bilmes and Kirchhoff, 2003] Bilmes, J. and Kirchhoff, K. (2003). Factored language models and generalized parallel backoff. In *Proceedings of HLT/NACCL*, pages 4–6.

[Chan et al., 1995] Chan, D., Fourcin, A., Lamel, L., and al. (1995). EUROM - a spoken langage resource for the EU. In *Proceedings of Eurospeech*, pages 867–870, Madrid, Spain.

[Charniak, 2001] Charniak, E. (2001). Immediate-head parsing for language models. In *Meeting of the Association for Computational Linguistics*, pages 116–123.

[Chase et al., 1994] Chase, L., Rosenfeld, R., and Ward, W. (1994). Error-responsive modifications to speech recognizers: negative n-grams. In *Proceedings of ICSLP*, pages 827–830, Yokohama, Japan.

[Chelba, 2006] Chelba, C. (2006). Acoustic sensitive language model perplexity for automatic speech recognition. http://snowbird.djvuzone.org/abstracts/001.pdf.

[Chelba and Jelinek, 2000] Chelba, C. and Jelinek, F. (2000). Structured language modeling. *Computer Speech and Language*, 14(4):283–332.

[Chen and Goodman, 1998] Chen, S. F. and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Center for Research in Computing Technology, Harvard University.

[Creutz and Lagus, 2005] Creutz, M. and Lagus, K. (2005). Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor. Technical Report A81, Helsinki University of Technology. URL:http://www.cis.hut.fi/projects/morpho/.

[Eek, 1974] Eek, A. (1974). Observations on the duration of some word structures: I. In *Estonian Papers in Phonetics*, pages 18–32.

[Eek and Meister, 1997] Eek, A. and Meister, E. (1997). Simple perception experiments in Estonian word prosody: foot structure vs. segmental quantity. In *Estonian Prosody: Papers from a Symposium*, pages 71–99, Tallinn, Estonia.

[Eek and Meister, 1999] Eek, A. and Meister, E. (1999). Estonian speech in the BABEL multi-language database: Phonetic-phonological problems revealed in the text corpus. In *Proceedings of LP'98. Vol II.*, pages 529–546.

[Ehala, 2006] Ehala, M. (2006). The Word Order of Estonian: Implications to Universal Language. *Journal of Universal Language*, (7):49–89.

[Engstrand and Krull, 1994] Engstrand, O. and Krull, D. (1994). Durational correlates of quantity in Swedish, Finnish and Estonian: Cross language evidence for a theory of adaptive dispersion. *Phonetica*, 51:80–91.

[Fiscus, 1997] Fiscus, J. (1997). A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (ROVER). In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–352, Santa Barbara, CA, USA.

[Furui, 1986] Furui, S. (1986). Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):52–59.

[Geutner et al., 1998] Geutner, P., Finke, M., and Scheytt, P. (1998). Adaptive vocabularies for transcribing multilingual broadcast news. In *Proceedings of ICASSP 1998*, Seattle, Washington.

[Gokcen and Gokcen, 1997] Gokcen, S. and Gokcen, J. (1997). A multilingual phoneme and model set: toward a universal base forautomatic speech recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 599–605, Santa Barbara, CA, USA.

[Good, 1953] Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3,4):237–264.

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

[Huang et al., 2001] Huang, X., Acero, A., Hon, H.-W., and Reddy, R. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR; 1st edition.

[Jelinek, 1989] Jelinek, F. (1989). Self-organized language modeling for speech recognition. In Waibel, A. and Lee, K.-F., editors, *Readings in Speech Recognition*. Morgan Kaufmann.

[Jelinek et al., 1992] Jelinek, F., Mercer, R., and Roukos, S. (1992). Principles of lexical language modeling for speech recognition. In Furui, S. and Sondhi, M. M., editors, *Advances in Speech Signal Processing*, pages 651–699. Marcel Dekker, Inc.

[Kaalep, 1998a] Kaalep, H.-J. (1998a). Kas vale meetodiga õiged tulemused? Statistikale tuginev eesti keele morfoloogiline ühestamine. *Keel ja Kirjandus*, (1):30–38.

[Kaalep, 1998b] Kaalep, H.-J. (1998b). Tekstikorpuse abil loodud eesti keele morfoloogiaanalüsaator. *Keel ja Kirjandus*, (1):22–29.

[Kaalep and Muischnek, 2005] Kaalep, H.-J. and Muischnek, K. (2005). The corpora of Estonian at the University of Tartu: the current situation. In *The Second Baltic Conference on Human Language Technologies : Proceedings*, pages 267–272, Tallinn, Estonia.

[Kaalep and Vaino, 2001] Kaalep, H.-J. and Vaino, T. (2001). Complete morphological analysis in the linguist's toolbox. In *Congressus Nonus Internationalis Fenno-Ugristarum Pars V*, pages 9–16, Tartu, Estonia.

[Kanis et al., 2005] Kanis, J., Zelinka, J., and Müller, L. (2005). Automatic numbers normalization in inflectional languages. In *Proceedings of SPECOM 2005*, pages 663–666, Patras, Greece.

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

[Killer, 2003] Killer, M. (2003). Grapheme based speech recognition. Master's thesis, Carnegie Mellon University.

[Kim et al., 2005] Kim, D., Chan, H., Evermann, G., Gales, M., Mrva, D., Sim, K., and Woodland, P. (2005). Development of the CU-HTK 2004 broadcast

news transcription systems. In *Proceedings of ICASSP*, volume 1, pages 861–864.

[Kirchhoff et al., 2002] Kirchhoff, K., Bilmes, J., Henderson, J., Schwartz, R., Noamany, M., Schone, P., Ji, G., Das, S., Egan, M., He, F., Vergyri, D., Liu, D., and Duta, N. (2002). Novel speech recognition models for Arabic. Technical report, Johns Hopkins University.

[Kirchhoff et al., 2006] Kirchhoff, K., Vergyri, D., Bilmes, J., Duh, K., and Stolcke, A. (2006). Morphology-based language modeling for conversational Arabic speech recognition. *Computer Speech & Language*, 20:589–608.

[Kneser and Ney, 1993] Kneser, R. and Ney, H. (1993). Improved clustering techniques for class-based statistical language modelling. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 973–976.

[Kneser and Ney, 1995] Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proceedings of ICASSP*, volume 1, pages 181–184.

[Kraut, 2000] Kraut, E. (2000). *Eesti keele hääldamine*. TEA Kirjastus.

[Krull, 1991] Krull, D. (1991). Stability in some Estonian duration relations. In *PERILUS*, number 13, pages 57–60. Institute of Linguistics, University of Stockholm.

[Kuhn and Ojamaa, 1989] Kuhn, G. M. and Ojamaa, K. (1989). Scores for Connected Recognition of Words Differing in Distinctive Quantity. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):1009–1019.

[Künnap, 1992] Künnap, E. (1992). *Eesti kõnekeele automaatne tuvastamine ja süntees : aruanne*. Eesti Teaduste Akadeemia Küberneetika Instituut, Tallinn.

[Kurimo et al., 2006] Kurimo, M., Puurula, A., Arisoy, E., Siivola, V., Hirsimäki, T., Pylkkönen, J., Alumäe, T., and Saraclar, M. (2006). Unlimited vocabulary speech recognition for agglutinative languages. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, HLT-NAACL 2006 : Proceedings of the Main Conference, June 4-9, 2006, New York, USA*, pages 487–494.

[Kwon and Park, 2003] Kwon, O.-W. and Park, J. (2003). Korean large vocabulary continuous speech recognition with morpheme-based recognition units. *Speech Communication*, 39:287–300.

[Lamel et al., 2006] Lamel, L., Gauvain, J.-L., Adda, G., Barras, C., Bilinski, E., Galibert, O., Pujol, A., Schwenk, H., and Zhu, X. (2006). The LIMSI 2006

TC-STAR transcription systems. In *TC-STAR Workshop on Speech-to-Speech Translation*, pages 123–128, Barcelona, Spain.

[Lamel et al., 1996] Lamel, L. F., Adda-Decker, M., Gauvain, J. L., and Adda, G. (1996). Spoken Language Processing in a Multilingual Context. In *Proceedings of ICSLP*, volume 4, pages 2203–2206, Philadelphia, PA, USA.

[Lamere et al., 2003] Lamere, P., Kwok, P., Walker, W., Gouvea, E., Singh, R., Raj, B., and Wolf, P. (2003). Design of the CMU Sphinx-4 decoder. In *Proceedings of Eurospeech*, Geneva, Switzerland.

[Lee et al., 2001] Lee, A., Kawahara, T., and Shikano, K. (2001). Julius — an open source real-time large vocabulary recognition engine. In *Proceedings of Eurospeech*, pages 1691–1694, Aalborg, Denmark.

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

[Lehiste, 1960] Lehiste, I. (1960). Segmental and syllabic quantity in Estonian. In *American Studies in Uralic Linguistics*, number 1, pages 21–82. Indiana Iniversity.

[Lehiste, 1966] Lehiste, I. (1966). Consonant quantity and phonological units in Estonian. *Indiana University Publications, Uralic and Altaic Series*, 65.

[Lihan et al., 2006] Lihan, S., Juhár, J., and Čižmár, A. (2006). Comparison of Slovak and Czech Speech Recognition Based on Grapheme and Phoneme Acoustic Models. In *Proceedings of ICSLP*, pages 149–152, Pittsburgh, USA.

[Liiv, 1961] Liiv, G. (1961). Eesti keele kolme vältusastme vokaalide kestus ja meloodiatüübid. *Keel ja Kirjandus*, (7):412–424.

[Lindström, 2000] Lindström, L. (2000). Narratiiv ja selle sõnajärg. *Keel ja Kirjandus*, (3):190–200.

[Mast et al., 1996] Mast, M., Kompe, R., Harbeck, S., Kiessling, A., and Warnke, V. (1996). Dialog act classification with the help of prosody. In *Proceedings of ICSLP*, volume 3, pages 1732–1735, Philadelphia, PA, USA.

[Maucec et al., 2003] Maucec, M. S., Rotovnik, T., and Zemljak, M. (2003). Modelling highly inflected Slovenian language. *International Journal of Speech Technology*, (6):245–257.

[McTait and Adda-Decker, 2003] McTait, K. and Adda-Decker, M. (2003). The 300k LIMSI German Broadcast News Transcription System. In *Proceeedings of Eurospeech*, Geneva, Switzerland.

[Meister, 2001] Meister, E. (2001). Towards speech recognition in Estonian. 21. Fonetiikan Päivät, Turku 4.-5.1.2001. *Publications of the Department of Finnish and General Linguistics of the University of Turku (eds. S.Ojala, J.Tuomainen)*, pages 59–70.

[Meister et al., 2002] Meister, E., Lasn, J., and Meister, L. (2002). Estonian SpeechDat: a project in progress. In *Fonetiikan Päivät 2002 — The Phonetics Symposium 2002*, pages 21–26.

[Meister et al., 2003] Meister, E., Lasn, J., and Meister, L. (2003). Development of the Estonian SpeechDat-like database. In *Proceedings of Eurospeech*, volume 2, pages 1601–1604, Geneva, Switzerland.

[Meister et al., 2004] Meister, E., Lasn, J., and Meister, L. (2004). Estonian SpeechDat completed. In Seppänen, T., Suomi, K., and Toivanen, J., editors, *Fonetikkan Päivät 2004*, pages 61–64, Oulu, Finland.

[Meister et al., 2001] Meister, E., Lobanov, B., Vahisalu, R., Levkovskaya, T., Kisialou, V., Tatter, P., and Lasn, J. (2001). Spoken dialogue system for mobile parking. In *Proceedings of the International Workshop SPEECH and COMPUTER (SPECOM 2001)*, pages 123–126, Moscow, Russia.

[Ney et al., 1994] Ney, H., Essen, U., and Kneser, R. (1994). On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1–28.

[Nouza et al., 2005] Nouza, J., Žďánský, J., David, P., Červa, P., Kolorenč, J., and Nejedlová, D. (2005). Fully automated system for Czech spoken broadcast transcription with very large (300k+) lexicon. In *Proceedings of Eurospeech*, pages 1681–1684, Lisbon, Portugal.

[Ohtsuki et al., 1999] Ohtsuki, K., Matsuoka, T., Mori, T., Yoshida, K., Taguchi, Y., Furui, S., and Shirai, K. (1999). Japanese large-vocabulary continuous-speech recognition using a newspaper corpus and broadcast news. *Speech Communication*, 28:155–166.

[Parandekar and Kirchhoff, 2003] Parandekar, S. and Kirchhoff, K. (2003). Multi-stream language identification using data-driven dependency selection. In *Proceedings of ICASSP*, volume 1, pages 28–31.

[Pellom, 2001] Pellom, B. (2001). SONIC: The University of Colorado continuous speech recognizer. Technical Report TR-CSLR-2001-01, University of Colorado, Boulder, Colorado, USA.

[Press et al., 1988] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press.

[Rabiner and Juang, 1986] Rabiner, L. and Juang, B. (1986). An introduction to hidden markov models (speech recognition and processing). *IEEE ASSP Magazine*, 3:4–16.

[Ramabhadran et al., 2006] Ramabhadran, B., Siohan, O., Mangu, L., Zweig, G., Westphal, M., Schulz, H., and Soneiro, A. (2006). The IBM 2006 speech transcription system for European parliamentary speeches. In *TC-STAR Workshop on Speech-to-Speech Translation*, pages 111–116, Barcelona, Spain.

[Remmel, 1963] Remmel, N. (1963). Sõnajärjestus eesti lauses. Deskriptiivne käsitlus. In *Eesti keele süntaksi küsimusi*, pages 216–271. Eesti Riiklik Kirjastus.

[Rosenfeld, 1995] Rosenfeld, R. (1995). Optimizing lexical and N-gram coverage via judicious use of linguistic data. In *Proceedings of Eurospeech*, pages 1763–1766, Madrid, Spain.

[Schukat-Talamazzini, 1995] Schukat-Talamazzini, E. (1995). *Automatische Spracherkennung - Grundlagen, statistische Modelle und effiziente Algorithmen*. Vieweg Verlag, Braunschweig.

[Schwenk and Gauvain, 2005] Schwenk, H. and Gauvain, J.-L. (2005). Building Continuous Space Language Models for Transcribing European Languages. In *Proceedings of Eurospeech*, Lisbon, Portugal.

[Shannon, 1948] Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27:379–423, 623–656.

[Shannon, 1953] Shannon, C. E. (1953). Communication theory: Exposition of fundamentals. *IEEE Transactions on Information Theory*, 1:44–47.

[Shriberg et al., 1997] Shriberg, E., Bates, R., and Stolcke, A. (1997). A prosody-only decision-tree model for disfluency detection. In *Proceedings of Eurospeech*, pages 2383–2386, Rhodes, Greece.

[Shriberg et al., 2000] Shriberg, E., Stolcke, A., Hakkani-Tür, D., and Tür, G. (2000). Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32(1-2):127–154.

[Siivola et al., 2003] Siivola, V., Hirsimäki, T., Creutz, M., and Kurimo, M. (2003). Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner. In *Proceedings of Eurospeech*, Geneva, Switzerland.

[Singh et al., 1999] Singh, R., Raj, B., and Stern, R. M. (1999). Automatic Clustering And Generation Of Contextual Questions For Tied States In Hidden Markov Models. In *Proceedings of ICASSP*, volume 1, pages 117–120.

[Stolcke, 2002] Stolcke, A. (2002). SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP 2002*, volume 2, pages 901–904, Denver, USA.

[Stolcke et al., 1998] Stolcke, A., Shriberg, E., Bates, R., Ostendorf, M., Hakkani, D., Plauche, M., Tur, G., and Lu, Y. (1998). Automatic detection of sentence boundaries and disfluencies based on recognized words. In *Proceedings of ICSLP*, volume 5, pages 2247–2250, Sydney, Australia.

[Sutrop, 2004] Sutrop, U. (2004). Estonian language. http://www.einst.ee/failid/eestikeel.web_1.pdf.

[Szarvas and Furui, 2003] Szarvas, M. and Furui, S. (2003). Evaluation of the stochastic morphosyntactic language model on a one million word Hungarian dictation task. In *Proceedings of Eurospeech*, Geneva, Switzerland.

[Tael, 1988] Tael, K. (1988). *Sõnajärjemallid eesti keeles (võrrelduna soome keelega)*. TA Keele ja Kirjanduse Instituut, Tallinn, Estonia.

[Uebel and Woodland, 1999] Uebel, L. and Woodland, P. (1999). An investigation into vocal tract length normalisation. In *Proceedings Eurospeech*, volume 6, pages 2527–2530, Budapest, Hungary.

[Venkataraman and Wang, 2003] Venkataraman, A. and Wang, W. (2003). Techniques for effective vocabulary selection. In *Proceedings of Eurospeech*, Geneva, Switzerland.

[Whittaker, 2000] Whittaker, E. (2000). *Statistical Language Modelling for Automatic Speech Recognition of Russian and English*. PhD thesis, University of Cambridge.

[Woodland et al., 1995] Woodland, P., Leggetter, C., Odell, J., Valtchev, V., and Young, S. (1995). The development of the 1994 HTK large vocabulary speech recognition system. In *Proceedings of ICASSP*, volume 1, pages 73–76.

[Young et al., 2003] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2003). The HTK Book (for HTK Version 3.2). http://htk.eng.cam.ac.uk.

[Young et al., 1994] Young, S., Odell, J., and Woodland, P. (1994). Tree-based state tying for high accuracy acoustic modelling. In *ARPA Workshop on Human Language Technology*, pages 307–312.

[Young et al., 1989] Young, S., Russell, N., and Thornton, J. (1989). Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems. Technical report, Cambridge University Engineering Department.

Appendix A

Phoneset mapping

The following mapping was used in the initial alignment of training sentences when training acoustic models using the SONIC toolkit.

IPA symbol	In Estonian phoneset	Sample	In English phoneset	Sample
ɑ	a	kabe	AH	but
æ	ae	käbi	AE	mad
e	e	tema	EH	bed
f	f	aferist	F	friend
h	h	hobune	HH	had
i	i	pime	IH	bitter
j	j	kaja	IH	bitter
g̊	k	pagar	GD	mug
k	K	pika, pikka	KD	talk
l, lʲ	l	lina	L	listen
m	m	mina	M	manager
n, nʲ	n	nina	N	nancy
o	o	oma	AO	for
ø	oe	lõmitama	AX	alone
ɤ	ou	kõmin	OW	cone
b̥	p	laba	BD	tab
p	P	tapa, tappa	P	pop
r	r	tara	R	red
s, sʲ	s	kask	S	sonic
ʃ	sh	kašelott, garaaž	SH	show
d̥, d̥ʲ	t	kade	DD	had
t, tʲ	T	kate, katte	T	tot
u	u	juhe	UW	moon
y	ue	lühis	UW	moon
v	v	kava	V	very

Appendix B

Sample recognition results

This appendix lists sample recognition results. The results are taken from the test set after applying the two-pass recognition technique as described in section 6.3.4. Every fifth sentence from the 320 sentence set is listed.

The recognized sentences (starting with *HYP:*) are aligned with reference sentences (starting with *REF:*). The identifier of the sentence uses the format `<sex><speaker id>_<session id>_<sentence id>`, where *m* stands for male speakers and *n* for female speakers. Incorrectly recognized words are written in uppercase. Asterisks are used to denote insertion and deletion errors.

```
id: (m50004_3_0037)
REF:  RAMBUJEE  on prantsusmaal aretatud villa ja ****
LIHALAMMAS
HYP:  HAMBULISE on prantsusmaal aretatud villa ja LIHA LAMAS

id: (m50015_648_0015)
REF:  mu naaber kutsus talgutele kõik oma sõbrad ja
sugulased **
HYP:  mu naaber kutsus talgutele kõik oma sõbrad ja
sugulased DA

id: (m50015_648_0045)
REF:  poiss SÖÖTIS ebatäpselt ja RÜNNAKUD luhtusid
HYP:  poiss SÕITIS ebatäpselt ja RÜNNAKUT luhtusid

id: (m50089_841_0032)
REF:  TURUL ja KLUBIL on sarnased funktsioonid
HYP:  TUUL  ja KLUBI  on sarnased funktsioonid

id: (m50103_20_0011)
REF:  ***** ELEEGIA on LÜHIKE POEETILISELT
rahvusromantiline PUHANG NAGU RESÜMEE pärast PIKA  reisi
LÕPPU
HYP:  ELIIT JA      on RIIKE  POLIITILISELT
rahvusromantiline PUHATA KURI SURVE   pärast PIKKA reisi
LÕPP

id: (m50103_20_0043)
```

```
REF:  PLÖRÖÖSID on ********* *** ***** PEALETÕM
PEALEÕMMELDAVAD LEINAPAELAD RIIDEIL JA    LEINAÄÄRIS
KIRJAÜMBRIKUL
HYP:  KÕRRESID  on PEALETUNG ÜHE PEALE ÕMMELDAVAD PIINAB
     ALATI       TEISE   LINNA TA         RISKIBJAID

id: (m50134_692_0020)
REF:  tanklaev mille pardal oli toornafta sõitis madalikule
**** KAGURANNIKU lähedal
HYP:  tanklaev mille pardal oli toornafta sõitis madalikule
KOGU RANNIKU     lähedal

id: (m50134_692_0049)
REF:  poiss sai sünnipäevaks suure ilusa troska kolm HOBUST
EES    JA    vene kutsar PUKI  PEAL
HYP:  poiss sai sünnipäevaks suure ilusa troska kolm ******
LOOBUS TEISE vene kutsar PUKKI VEAB

id: (m50172_381_0037)
REF:  kaljude kohal LENDLEVAID kotkaid vaadates MÕISTAD kui
võimsad linnud NEED ON
HYP:  kaljude kohal LENDAVAID  kotkaid vaadates MÕISTAB kui
võimsad linnud **** TOLM

id: (m50279_503_0015)
REF:  SEENE KAUSJASSE      kübarasse oli kogunenud vihmavett
HYP:  ***** SEENEKAUSSESSE kübarasse oli kogunenud vihmavett

id: (m50279_503_0045)
REF:  VEEGA ristimine oli patukahetsuse ja PATTUDE
ANDEKSANDMISE OTSIMISE             sümbol
HYP:  SEEGA ristimine oli patukahetsuse ja ******* PATUD
    ANDEKSANDMISOTSIMISE sümbol

id: (m50353_620_0032)
REF:  RIIA ja vilniuse BÖRSIDEGA VÕRRELDES   on tallinna
börsi KAPITALINÕUDED SUURUSJÄRU SUURUSJÄRGU         VÕRRA
KÕRGEMAD
HYP:  **** ja vilniuse BÖRSI     TEGEVAJADES on tallinna
börsi ************** ********** KAPITALINÕUDEDVASSE JÄRGU
VAJADUST

id: (m50425_485_0011)
REF:  ***** LUPJASIME happelist mulda ET taimede
kasvutingimusi parandada
HYP:  MITTE SINNA     happelist mulda ** taimede
kasvutingimusi parandada

id: (m50425_485_0043)
REF:  ***** SEGIDILJA on andaluusiast pärinev hispaania
rahvalaul tants
HYP:  SEEGI SIIA      on andaluusiast pärinev hispaania
rahvalaul tants

id: (m50595_279_0020)
REF:  ETTEVÕTJA jaoks on tähtis adekvaatne informatsioon
turul toimuvast
HYP:  ETTEVÕTTE jaoks on tähtis adekvaatne informatsioon
turul toimuvast

id: (m50595_279_0049)
```

```
REF:  MAJAKESES EI   OLE MADRATSIT TEKKI     PATJA LINADEST
EI       maksa rääkidagi
HYP:  ********* MAJA MIS JALA      MADRATSID TEKID PADJAD
PIINADES maksa rääkidagi

id: (m50717_467_0037)
REF:  MILJARDÄRIL seisis maja taga ******* PARKIMISPLATSIL
KOMPLEKTNE kogu fordi ****** AUTOTEHASE autosid
HYP:  MILJARDÄRI  seisis maja taga PARKLAS PLATSIL
KOMPLEKTE  kogu fordi AUTODE PEALT      autosid

id: (m50718_421_0015)
REF:  meie vaatepunktist oli tema artikkel puhas plagiaat
HYP:  meie vaatepunktist oli tema artikkel puhas plagiaat

id: (m50718_421_0045)
REF:  kaubavahetus suurenes mitmekordselt
HYP:  kaubavahetus suurenes mitmekordselt

id: (m50734_672_0032)
REF:  SELLEST KATSEST EI    TULNUD midagi välja
HYP:  ******* SELLE   KATSE KIDUR  midagi välja

id: (m50863_1306_0011)
REF:  ENDISE  meierei SÜGAVAD KELDRID OLID lagunenud ja
räämas
HYP:  INGLISE meierei SÜGAVA  KELDRI  TULI lagunenud ja
räämas

id: (m50863_1306_0043)
REF:  kuusk loob JÕULUDE AJAL      erilise meeleolu
HYP:  kuusk loob ******* JÕULUAJAL erilise meeleolu

id: (m50915_690_0020)
REF:  milleks pealetükkiv FAMILIAARSUS MIS    SAATEJUHI
KÄEGA VAJUB nagu KOOREM MÄNGIJA ÕLULE
HYP:  milleks pealetükkiv OMA          JAATUS SAATE
KÕIGE KOJU  nagu ****** MÕND    JÕULU

id: (m51043_1218_0011)
REF:  see pole aga enam ei PRIMITIVISM EGA
populism vaid lausdemagoogia
HYP:  see pole aga enam ei TEE         PRIMITIVISMIGA
populism vaid lausdemagoogia

id: (m51043_1218_0043)
REF:  kuriteoohvrite toetusfondi algkapital tuleks projekti
kohaselt VALITSUSE reservfondist
HYP:  kuriteoohvrite toetusfondi algkapital tuleks projekti
kohaselt VALITSEV  reservfondist

id: (m51239_834_0020)
REF:  MÕNE  kantselei *** TÖÖÕHKKONDA ISELOOMUSTAB vaikiv
vaenulikkus
HYP:  MÕNED kantselei TÖÖ KONTORIS    ALUSTAB      vaikiv
vaenulikkus

id: (m51239_834_0049)
REF:  vana maja KORSTNAT EI SAANUDKI   ÄRA    PARANDADA
korsten tuli uuesti laduda
HYP:  vana maja ******** ** KORSTNASSE SAANUD KÄRATA
```

```
korsten tuli uuesti laduda

id: (m51365_1348_0037)
REF:  eile kutsuti häirekeskusest viis korda tulekahjustusi
likvideerima
HYP:  eile kutsuti häirekeskusest viis korda tulekahjustusi
likvideerima

id: (m51445_1049_0015)
REF:  mul polnud isu portsjon oli liiga suur
HYP:  mul polnud isu portsjon oli liiga suur

id: (m51445_1049_0045)
REF:  SANATOORIUM  suudab välja müüa pooled kohtadest
HYP:  SANATOORIUMI suudab välja müüa pooled kohtadest

id: (m51490_857_0032)
REF:  ARST seletas patsiendile ET ENTERIIT ON
peensoolepõletik
HYP:  AS   seletas patsiendile ** TEEN     TRIITON
peensoolepõletik

id: (m51629_4263_0011)
REF:  täringumänguks oli lauale PANDUD VÜRFEL PLIIATS
JA       paberileht
HYP:  täringumänguks oli lauale ****** ****** PANNUDLISELT
PLIIATSI paberileht

id: (m51629_4263_0043)
REF:  GRAFFITI ON      noorte spontaanne eneseväljendus
teisalt aga oma territooriumi TÄHISTAMINE
HYP:  ******** GRAFITI noorte spontaanne eneseväljendus
teisalt aga oma territooriumi TEIST

id: (m51651_1184_0020)
REF:  on ütlemata tore vaadata kuidas pisipõnnid kõrgest
TRAMPLIINIST alla tuiskavad
HYP:  on ütlemata tore vaadata kuidas pisipõnnid kõrgest
RUTIINIST    alla tuiskavad

id: (m51651_1184_0049)
REF:  puhtaks pestud pisipoiss puges pidzhaamasse
HYP:  puhtaks pestud pisipoiss puges pidzhaamasse

id: (m51771_1382_0037)
REF:  kuid meenutades kuidas ta oli SUURTÜKIPAUGUST
EHMATANUD SAI    ADMIRAL ÄKKI kurjaks
HYP:  kuid meenutades kuidas ta oli PAUS            TEHA
 SAJANI AGA     LÄKI kurjaks

id: (m51777_4431_0015)
REF:  laulu looja oli huvitatud ainuautorsusest
HYP:  laulu looja oli huvitatud ainuautorsusest

id: (m51777_4431_0045)
REF:  kummalegi POJALE  tuleb anda õiglane ja võrdne
lähtepositsioon
HYP:  kummalegi POOLELE tuleb anda õiglane ja võrdne
lähtepositsioon

id: (m51812_1938_0032)
```

```
REF:  kõige rohkem MILJARDÄRE on ameerikas
HYP:  kõige rohkem MILJARDÄR  on ameerikas

id: (m51961_1946_0011)
REF:  rannaküla POISSE ÕPETATI juba NOOREST EAST poisse kui
meremärki AUSTAVALT suhtuma
HYP:  rannaküla POISS  LÕPETAS juba NOORES  EAS  poisse kui
meremärki AUSTATUD  suhtuma

id: (m51961_1946_0043)
REF:  PUTUKA KERE     OLI ÜLALT pruunikas või rohekaskollane
ALT VALKJAS   või hõbedane
HYP:  ****** PUTUKATE JA  LILLA pruunikas või rohekaskollane
*** ALFALIKES või hõbedane

id: (m52030_2183_0020)
REF:  mandoliini mängitakse erilise LIPITSAGA
HYP:  mandoliini mängitakse erilise LIIKIDEGA

id: (m52030_2183_0049)
REF:  PIPETT ON     hea vahend ROHU tilgutamiseks NINNA
HYP:  ****** PEETER hea vahend OHU  tilgutamiseks MINNA

id: (m52485_3105_0037)
REF:  sellest saab alguse LÜHIKE kirglik ja valuline
armastuslugu
HYP:  sellest saab alguse RÜÜTLI kirglik ja valuline
armastuslugu

id: (m52489_4060_0015)
REF:  kui NÄED ET   POISIL loksub adrenaliin silmis tuleb
teda mõistusele kutsuda
HYP:  kui **** NÕEL POISI  loksub adrenaliin silmis tuleb
teda mõistusele kutsuda

id: (m52489_4060_0045)
REF:  ägedalt SPURTIMA
HYP:  ägedalt SPORTIMA

id: (n50098_449_0032)
REF:  möödunud kevadel seadsin endale eesmärgiks selles
ametis SÜGISENI vastu pidada
HYP:  möödunud kevadel seadsin endale eesmärgiks selles
ametis SINISENI vastu pidada

id: (n50477_584_0011)
REF:  ***** PELARGOON SOBIB tugevamale inimesele MONSTERA
JÕULISEMALE MATERIALISTILE ROHTLIILIA         NÄRVILISELE
        INIMESELE
HYP:  PEAME KROON     SOBI  tugevamale inimesele MA
TÄNA        JÕULISE        MALEMATERJALISTILE
ROHTLIILIANÄRVILISTELE INIMES

id: (n50477_584_0043)
REF:  ***** TARMO     JUURDLES ALATASA MAAILMA asjade üle
HYP:  TARMU JUURDLEJA ANDA     SAMA    ILMA    asjade üle

id: (n50494_270_0020)
REF:  on ütlemata tore vaadata kuidas pisipõnnid KÕRGEST
trampliinist alla tuiskavad
HYP:  on ütlemata tore vaadata kuidas pisipõnnid KÕRGES
```

```
trampliinist alla tuiskavad

id: (n50494_270_0049)
REF:  PUHTAKS PESTUD pisipoiss puges pidzhaamasse
HYP:  KOHTAB  TEHTUD pisipoiss puges pidzhaamasse

id: (n50625_531_0037)
REF:  AVA AKORD    ei olnud õnnestunud valik
HYP:  *** KANAKORD ei olnud õnnestunud valik

id: (n50873_875_0015)
REF:  eesti ajal sai ta asunikutalu KEISRI AJAL oli ta aga
olnud kõigest **** MÕISA teomees
HYP:  eesti ajal sai ta asunikutalu SIIS   SEAL oli ta aga
olnud kõigest MÕNI ISA   teomees

id: (n50873_875_0045)
REF:  ta väljus ja virutas ukse tagantkätt nii hoogsalt
kinni et ***** PORTJÄÄR paisus nagu ***** PURI TUULES
HYP:  ta väljus ja virutas ukse tagantkätt nii hoogsalt
kinni et SPORT JÄÄDA    paisus nagu PUURI II   TUULEST

id: (n51159_1720_0032)
REF:  IDIOOM  on ********************** LIIK
FRASEOLOOGILIS FRASEOLOGISME keeles **** JUURDUNUD omapärane
kõnekäänd
HYP:  IDIOOMI on KICKFRASEOLOOGIALISPÄEV ARHEOLOOG KISS
     ME            keeles JUUR TULUTU    omapärane kõnekäänd

id: (n51196_1922_0011)
REF:  paleoliitikumi AJAL valmistas ÜRGINIMENE TÖÖRIISTU
RAIUMISE teel kuna LIHVIMINE OLI VEEL          tundmatu
HYP:  paleoliitikumi HÄÄL valmistas MÜRGINE    TÖÖLISTE
RAJAMISE teel kuna ********* *** LIHVIMINELINE tundmatu

id: (n51196_1922_0043)
REF:  tuntud sprinter SAI  END   OLÜMPIAVÕITJANA tunda vaid
ühe ÖÖPÄEVA sest jäi vahele positiivse DOPINGUPROOVIGA
HYP:  tuntud sprinter ENDA NENDE VÕITJAD         tunda vaid
ühe PÄEVA   sest jäi vahele positiivse DOPINGUPROOVINA

id: (n51203_860_0020)
REF:  ENNE munadepühi varus ema alati ****** PASHAKS MÕELDUD
KOHUPIIMA
HYP:  ENE  munadepühi varus ema alati KAASAS MAJA    PEALT
KOHUPIIMALT

id: (n51203_860_0049)
REF:  NAGU  TRANSSI langenuna soigus ÕNNETUKE pead
vangutades ***** NURGAS
HYP:  MÄNGU TANTSI  langenuna soigus ÕNNETUTE pead
vangutades NURGA ALT

id: (n51299_4357_0037)
REF:  allkiri identifitseerib maksja isiku ja panga
HYP:  allkiri identifitseerib maksja isiku ja panga

id: (n51321_939_0015)
REF:  lõppes esimene poolfinaal
HYP:  lõppes esimene poolfinaal
```

```
id: (n51321_939_0045)
REF:  mõni lihtsam natüürmort sobib väga hästi KÖÖKI
HYP:  mõni lihtsam natüürmort sobib väga hästi KEEGI

id: (n52159_4336_0032)
REF:  kas SUL on ka pedagoogikaõppejõudude seas MÕNI lemmik
HYP:  kas *** on ka pedagoogikaõppejõudude seas MÕNE lemmik
```

Appendix C

Curriculum Vitae

1. Personal Data

Name:	Tanel Alumäe
Date and place of birth:	29.05.1976, Tallinn
Citizenship	Estonian
Marital status:	single
Children	-

2. Contact Data

Address:	A. Kapi 3-4, 10136, Tallinn, Estonia
Phone:	+372 56 916761
E-mail:	tanel.alumae@phon.ioc.ee

3. Education

Educational Institution	*Graduation time*	*Speciality / grade*
Tallinn Technical University	2002	Information Technology / Master of Science in Engineering
Tallinn Technical University	1999	Information Technology / Graduate Engineer

4. Language Skills (basic, intermediate or high level)

Language	*Level*
Estonian	Mother Tongue
English	High Level
German	High Level

5. Special Courses

Course and time	*Educational institution or organisation*
Summer School on Variation In Speech Production and Speech Perception, Palmse, Estonia, August 2005	NorFA
16th European Summer School in Logic, Language and Information, Nancy, France, August 2004	European Association for Logic, Language and Information
Winter School on Speech Production Modelling, Helsinki, January 2004	Graduate School of Language Technology in Finland
International Masters Program in Computational Engineering, one year	University of Erlangen-Nuremberg, Germany

6. Professional employment

Period	*Institution*	*Position*
09/2003 -	Institute of Cybernetics at Tallinn University of Technology	Researcher
10/2000 - 11/2006	AS Aqris Software	Senior Software Developer

8. Scientific Work

Kurimo, M., Puurula, A., Arisoy, E., Siivola, V., Hirsimaki, T., Pylkkonen, J., Alumäe, T. and Saraclar, M. Unlimited vocabulary speech recognition for agglutinative languages. In Human Language Technology, Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2006. New York, USA, June 5-7, 2006, pp. 487–494.

Alumäe, T. Sentence-adapted Factored Language Model for Transcribing Estonian Speech. In Proceedings of ICASSP 2006. Toulouse, France, vol. 1, pp. 429–432.

Alumäe, T. Using Adaptive Stochastic Morphosyntactic Language Model for Two-pass Large Vocabulary Estonian Speech Recognition. Proceedings of the 10th Internation Conference SPEECH and COMPUTER, 17 - 19 October 2005, Patras, Greece, pp. 515–518.

Alumäe, T. Phonological and morphological modeling in large vocabulary continuous Estonian speech recognition system. The Second Baltic Conference on Human Language Technologies : Proceedings, April 4-5, 2005, Tallinn, Estonia / Eds. M. Langements, P. Penjam. Tallinn : Institute of Cybernetics (Tallinn University of Technology); Institute of the Estonian Language, 2005, pp. 89–94.

Alumäe, T. Estonian Speech Recognition Experiments using the SpeechDat-Like Database. Fonetiikan Päivät 2004 / Eds. T. Seppänen, K. Suomi, J. Toivanen. Oulu, 2005, pp. 65–68.

Meister, E,. Alumäe. T. Recent Advances in Estonian Spoken Language Technology. Baltic IT&T Review, vol. 33, pp. 66–69.

Alumäe, T. Large Vocabulary Continuous Speech Recognition for Estonian Using Morpheme Classes. Proceedings of Interspeech 2004 - ICSLP, pp. 389–392

Alumäe, T., Võhandu, L. Limited-vocabulary Estonian continuous speech recognition system using hidden Markov models. Informatica, vol. 15, No. 3, 2004, pp. 303–314

Alumäe, T. Large Vocabulary Continuous Speech Recognition for Estonian Using Morphemes and Classes. Proceeding of the 7th International Conference, TSD 2004, lk. 245–252

Alumäe, T. Large Vocabulary Continuous Speech Recognition for Estonian Using Morphemes and Classes. Proceeding of the First Baltic Conference: Human Language Technologies - The Baltic Perspective, 2004, pp. 166–169

Additionally, three publications in Estonian (see CV in Estonian for more information)

9. Theses Accomplished and Defended

Graduate Engineer Thesis (1999): Design and implementation of an OLAP system.

M. Sc. Thesis (2002): Limited vocabulary Estonian speech recognition.

10. Research Interests

Speech processing, language modelling, statistical methods, machine learning

11. Other Research Projects

-

Signature: Date:

1. Isikuandmed

Ees- ja perekonnanimi:	Tanel Alumäe
Sünniaeg ja -koht:	29.05.1976, Tallinn
Kodakondsus	Eesti
Perekonnaseis:	vallaline
Lapsed	-

2. Kontaktandmed

Aadress:	A. Kapi 3-4, 10136, Tallinn, Estonia
Telefon:	+372 56 916761
E-posti aadress:	tanel.alumae@phon.ioc.ee

3. Hariduskäik

Õppeasutus (nimetus lõpetamise ajal)	*Lõpetamise aeg*	*Haridus (eriala / kraad)*
Tallinna Tehnikaülikool	2002	Infotehnoloogia / Tehnikateaduste magister
Tallinna Tehnikaülikool	1999	Arvuti- ja süsteemitehnika / Insener

4. Keelteoskus (alg-, kesk-, või kõrgtase)

Keel	*Tase*
Eesti	Emakeel
Inglise	Kõrgtase
Saksa	Kõrgtase

5. Täiendõpe

Kursus ja õppimise aeg	*Õppeasutuse või muu organisatsiooni nimetus*
Suvekool ”Variation In Speech Production and Speech Perception”, Palmse, Eesti, August 2005	NorFA
16th European Summer School in Logic, Language and Information, Nancy, France, August 2004	European Association for Logic, Language and Information
Winter School on Speech Production Modelling, Helsinki, January 2004	Graduate School of Language Technology in Finland
Rahvusvaheline magistriprogramm ”Computational Engineering”, 1 aasta	Erlangen-Nürnbergi ülikool, Saksamaa

6. Teenistuskäik

Töötamise aeg	*Ülikooli, teadusasutuse või muu organisatsiooni nimetus*	*Ametikoht*
09/2003 -	Tallinna Tehnikaülikooli Küberneetika Instituut	Teadur
10/2000 - 11/2006	AS Aqris Software	Tarkvaraarendaja

8. Teadustegevus

Kurimo, M., Puurula, A., Arisoy, E., Siivola, V., Hirsimaki, T., Pylkkonen, J., Alumäe, T. and Saraclar, M. Unlimited vocabulary speech recognition for agglutinative languages. In Human Language Technology, Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2006. New York, USA, June 5-7, 2006, lk. 487–494.

Alumäe, T. Sentence-adapted Factored Language Model for Transcribing Estonian Speech. In Proceedings of ICASSP 2006. Toulouse, France, vol. 1, lk. 429–432.

Alumäe, T. Using Adaptive Stochastic Morphosyntactic Language Model for Two-pass Large Vocabulary Estonian Speech Recognition. Proceedings of the 10th Internation Conference SPEECH and COMPUTER, 17 - 19 October 2005, Patras, Greece, lk. 515–518.

Alumäe, T. Phonological and morphological modeling in large vocabulary continuous Estonian speech recognition system. The Second Baltic Conference on Human Language Technologies : Proceedings, April 4-5, 2005, Tallinn, Estonia / Eds. M. Langements, P. Penjam. Tallinn : Institute of Cybernetics (Tallinn University of Technology); Institute of the Estonian Language, 2005, lk. 89–94.

Alumäe, T. Estonian Speech Recognition Experiments using the SpeechDat-Like Database. Fonetiikan Päivät 2004 / Eds. T. Seppänen, K. Suomi, J. Toivanen. Oulu, 2005, lk. 65–68.

Meister, E,. Alumäe, T. Recent Advances in Estonian Spoken Language Technology. Baltic IT&T Review, nr. 33, 2004, lk. 66–69.

Alumäe, T. Large Vocabulary Continuous Speech Recognition for Estonian Using Morpheme Classes. Proceedings of Interspeech 2004 - ICSLP, lk. 389–392

Alumäe, T., Kirt, T. Inimene on arvutile võõrkeel. Horisont 4/2004, lk. 42–44.

Alumäe, T., Võhandu, L. Limited-Vocabulary Estonian Continuous Speech Recognition System Using Hidden Markov Models. INFORMATICA, vol. 15, No. 3, 2004, lk. 303–314.

Alumäe, T. Large Vocabulary Continuous Speech Recognition for Estonian Using Morphemes and Classes. Proceeding of the 7th International Conference, TSD 2004, lk. 245–252

Alumäe, T. Eksperimendid eesti keele kõnetuvastussüsteemi loomisel. Tallinna Pedagoogikaülikooli eesti filoloogia osakonna toimetised 3. Toimiv keel II – Töid rakenduslingvistika alalt, 2004, lk. 23–36.

Alumäe, T. Eestikeelse kõne tuvastus: prototüübi loomine. Eesti Keele Instituudi toimetised 12. Toimiv keel I, 2003, lk. 34–49.

Alumäe, T., Võhandu, L. Piiratud ulatusega eestikeelne kõnetuvastus. Eesti Keele Instituudi toimetised nr 12. Toimiv keel I, 2003, lk. 50–52.

Alumäe, T. Varjatud Markovi mudelid. Arvutitehnika ja andmetöötlus, 4/2002, lk. 27–36.

9. Kaitstud lõputööd

Diplomitöö (1999): OLAP-süsteemi disain ja implementatsioon

Magistritöö (2002): Piiratud sõnavaraga eestikeelne kõnetuvastus

10. Teadustöö põhisuunad

Kõnetöötlus, keelemudelid, statistilised meetodid, masinõpe

11. Teised uurimisprojektid

–

Allkiri: Kuupäev: