Multilingual speech recognition
A posterior based approach

THÈSE N^{O} 5800 (2013)

PRÉSENTÉE LE 20 JUIN 2013

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

LABORATOIRE DE L'IDIAP

PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

David IMSENG

acceptée sur proposition du jury:

Prof. J.-Ph. Thiran, président du jury
Prof. H. Bourlard, directeur de thèse
Prof. N. Morgan, rapporteur
Prof. T. Schultz, rapporteur
Dr J.-M. Vesin, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2013

Fer mini Eltru.
To my parents.

Acknowledgements

Writing a thesis is like climbing a mountain. It was a long and demanding way. I am grateful to people for their support and advice along the way to finish my thesis. Climbing a mountain without a guide may be more dangerous than writing a thesis without a supervisor, but writing a thesis without a supervisor is hardly possible. I am indebted to my supervisor, Hervé Bourlard, for the guidance, the patience and the constructive and critical comments. I am also much obliged to Nelson Morgan for enabling me to do two internships at the International Computer Science Institute in Berkeley. I met many new people and had a lot of fruitful discussions during my time in Berkeley.

I would also like to thank many colleagues at Idiap Research Institute for their time and support. Thank you Phil, for all your help during the last four years. Your door was always open for informal chatting as well as serious discussions. Many thanks also go to Mathew, Petr, John, Gwénolé, Ramya, Milos, Alexandre, Raphaël and Holger. Furthermore, I would like to acknowledge colleagues at the International Computer Science Institute in Berkeley. Thank you Steve, Korbinian and Hari.

Finally, special thanks go to my parents Peter and Ruth, who allowed me to study what I like. I also express my gratitude to Erika and all my friends for their understanding and support.

Martigny, 11. April 2013 David Imseng

On the way to the top of the Mönch, 4,107 m above sea level.

Abstract

Modern automatic speech recognition (ASR) systems are based on parametric statistical models such as hidden Markov models (HMMs), exploiting 1) acoustic-phonetic models, which need to be trained on large amount of acoustic data, 2) a language model, which needs to be trained on large amount of text data and, finally, 3) a lexicon with phonetic transcription which requires linguistic expertise. Developing multilingual ASR systems, or systems that are robust to accents and dialects, is therefore a very challenging task for current state-of-the-art ASR systems.

In this thesis, we focus on investigating acoustic-phonetic modeling and lexical diversity across languages and databases, and assume that a language model is available. In our case, this is done in the context of hybrid HMM/MLP ASR, where the HMM emission probabilities are modeled as posterior probabilities of HMM states, conditioned on the acoustics, estimated at the output of a multilayer perceptron (MLP). We build upon a recently proposed acoustic modeling approach, referred to as KL-HMM, where posterior probabilities are directly used as acoustic features, and where the HMM states are directly parametrized by trained posterior probabilities. The set of HMM reference posteriors is then estimated by minimizing the Kullback–Leibler divergence between posterior features extracted from the training data and reference posteriors.

The proposed KL-HMM model is extensively developed and adapted to tackle several challenging problems related to multilingual ASR, including lexical diversity, stochastic phone space transformations, accented speech recognition and using multilingual data resources to boost monolingual systems. The efficiency of the proposed approach is demonstrated through theoretical and experimental comparisons with similar approaches such as probabilistic acoustic mapping, linear hidden networks and maximum a posteriori adaptation. Furthermore, KL-HMM is also compared with other posterior feature based ASR techniques such as Tandem and short-term spectral feature based approaches such as subspace Gaussian mixture models. The comparison reveals that the KL-HMM framework is a suitable alternative to conventional acoustic modeling techniques and seems to be preferable in low amount of data as well as phoneme set mismatch scenarios.

Keywords Multilingual speech recognition, multilingual acoustic modeling, posterior features, KL-HMM, non-native speech recognition, under-resourced languages

Zusammenfassung

Moderne automatische Spracherkennungssysteme basieren auf parametrischen statistischen Modellen, wie *hidden Markov Models (HMM)*, und bestehen aus drei Komponenten: 1) akustisch-phonetische Modelle, deren Berechnung grosse Mengen an akustischen Daten benötigt, 2) Sprachmodell, welches mit umfangreichen Textkorpora trainiert wird und, 3) Aussprachewörterbücher, die linguistisches Expertenwissen erfordern. Die Entwicklung von Systemen, welche auch akzentuierte Sprache und Dialekte korrekt verarbeiten, sowie multilinguale Spracherkennungssysteme, ist daher anspruchsvoll.

Diese Dissertation befasst sich mit der Erforschung akustisch-phonetischer Modelle sowie der lexikalischen Vielfalt der Aussprachewörterbücher über mehrere Sprachen und Datenbanken unter der Annahme, dass ein Sprachmodell verfügbar ist. Die präsentierte Forschung wird im Kontext von hybriden HMM/KNN Spracherkennern betrieben. Die Emissionswahrscheinlichkeiten des HMMs werden mit a-posteriori Wahrscheinlichkeiten der HMM Zustände, bedingt durch das akustische Signal und geschätzt von einem künstlichen neuronalen Netz (KNN), modelliert. Namentlich wird der vor kurzem eingeführte akustische Modellierungsansatz KL-HMM erforscht. KL-HMM benützt a-posteriori Wahrscheinlichkeiten direkt als akustische Merkmale (Posterior-Merkmale) und modelliert die HMM Zustände mittels trainierten a-posteriori Wahrscheinlichkeiten (Referenz-Posteriors). Diese Referenz-Posteriors können mittels Minimierung der Kullback–Leibler Divergenz zwischen Posterior-Merkmalen aus den Trainingsdaten und Referenz-Posteriors geschätzt werden.

Der KL-HMM Modellierungsansatz wird ausführlich entwickelt und angepasst um verschiedene anspruchsvolle Probleme zu bewältigen mit Bezug auf multilinguale Spracherkennung, lexikalische Vielfalt, stochastische Phonbereichstransformationen, akzentuierte Spracherkennung und Nutzung multilingualer Datensätze zur Verbesserung monolingualer Spracherkenner. Die Effizienz der eingebrachten Ansätze wird durch theoretische und experimentelle Vergleiche mit ähnlichen Verfahren, wie probabilistische akustische Zuordnung, lineare verborgene Netze und maximale a-posteriori Adaption, belegt. Des Weiteren wird KL-HMM mit anderen akustischen Modellierungsansätzen verglichen: Tandem, das auf Posterior-Merkmalen basiert, sowie Teilraum Gausssche Mischverteilungsmodelle (*subspace Gaussian mixture models*), welche auf spektralen Merkmalen basieren. Der Vergleich macht deutlich, dass KL-HMM eine geeignete Alternative zu konventionellen akustischen Modellierungsansätzen darstellt, und bei Szenarien mit limitiertem Datenmaterial oder Phonemesetdiskrepanz vorzuziehen ist.

Schlüsselwörter Multilinguale Spracherkennung, multilinguale akustische Modelle, Posterior-Merkmale, KL-HMM, Spracherkennung akzentuierter Sprache von Nicht-Muttersprachlern, Sprachen mit limitiertem Datenmaterial

Contents

List of Figures

List of Tables

List of Acronyms

ARPA advanced research projects agency

ASR automatic speech recognition

CGN corpus gesproken nederlands

CMU Carnegie Mellon University

CPDLC controller pilot data link communication

DTW dynamic time warping

EM expectation-maximization

GMM Gaussian mixture model

HMM hidden Markov model

HMM/GMM hidden Markov model based acoustic modeling based on generative Gaussian mixture models

HMM/MLP hidden Markov model based acoustic modeling based on discriminative multi-layer perceptrons

HTK hidden Markov model toolkit

IPA international phonetic alphabet

KL Kullback–Leibler

KL-HMM Kullback–Leibler divergence based hidden Markov model

LHN linear hidden network

LPC linear predictive coding

MAP maximum a posteriori

MDL minimum description length

MFCC mel-frequency cepstrum coefficients

MF-PLP mel-frequency PLP

ML-tag language-tagged acoustic modeling technique

MLLR maximum likelihood linear regression

MLP multilayer perceptron

PAM probabilistic acoustic mapping

PCA principal component analysis

PCM puls code modulation

PLP perceptual linear prediction

PPM probabilistic phone mapping

SAMPA speech assessment methods phonetic alphabet

SCHMM semi-continuous HMM

SGMM subspace Gaussian mixture model

TM template matching

TIMIT well-known acoustic-phonetic continuous speech corpus

UBM universal background model

WACC word accuracy

WAVE waveform audio file format

1 Introduction

I grew up in Valais, a bilingual canton of Switzerland. Valais is a valley, surrounded by mountains and is better known internationally for its ski resorts like Verbier and Zermatt with the Matterhorn, shown here on the right. For automatic speech recognition (ASR) research, Valais is interesting, because there are two different official languages, French and German. Furthermore, even within Valais, a region of about 5,000 square kilometers and a population of 300,000, there are many local accents and dialects, especially in the German speaking part. Actually, the German that is spoken in Valais is a group of dialects, also known as *Wallissertitsch* [Grichting, 2011], without standardized written form. The dialects differ a lot from the standard high German (Hochdeutsch), spoken in Germany, and are sometimes even difficult to understand for other Swiss German speakers. Close to the language border, Italy and French speaking Valais, people also use foreign words (loan words) in their dialect. In contrast to the shape of the Matterhorn, which is claimed to be one-of-a-kind, the language situation in Valais is interesting, but far from being unique. There are many regions in the world, where multiple languages are used in parallel and influence each other. This language mix leads to obvious difficulties, with many people working and even living in a non-native language, and involves numerous challenges for state-of-the-art ASR systems. One of the main goals of this thesis is to tackle some of the issues related to acoustic-phonetic modeling of such recordings.

Figure 1.1: The Matterhorn, a “classical” Swiss mountain landmark.

1.1 Multilingual speech recognition

State-of-the-art ASR systems typically use hidden Markov models (HMMs) and usually build on three components: acoustic-phonetic models, language model and a lexicon. If at least one of these components is multilingual, we refer to the whole system as a multilingual ASR system.

Multilingual language models are particularly useful when the speaker switches between languages (code-switching) or when the spoken language is unknown prior to decoding. Language models are normally trained on large amounts of text data. If text corpora from multiple languages are merged to estimate a multilingual language model, a language switch is in principle allowed at any time [Ward et al., 1998]. More restrictive approaches only allow language switches at common pause models [Weng et al., 1997]. Even though ASR systems with multilingual language models allow to implicitly identify the spoken language, if the spoken language is known a priori, usually the speech recognition performance is lower compared to ASR systems with monolingual language models [Fugen et al., 2003]. In this thesis, we assume that the language model is given, and we focus on improving acoustic and lexical models of monolingual systems, including crosslingual phone mapping and accented speech recognition.

In a similar vein, acoustic models can be trained on speech data from multiple languages. The main findings of multilingual acoustic modeling studies such as [Schultz and Waibel, 2001, Köhler, 2001], can be generalized as follows [Van Compernolle, 2001]:

- If there is enough training data, multilingual acoustic models perform worse than monolingual ones.
- The effect is more pronounced if the data from more diverse languages are merged during training.
- Such systems have a high practical value, especially when little or no data exists in a particular language.

In this thesis we expatiate upon the last point. More specifically, to model variability in the speech recordings, the acoustic models need to be trained on large amounts of acoustic data. Data collection involves large amounts of manual work, not only in time for the speakers to be recorded, but also for annotation of the subsequent recordings. Therefore, developing ASR systems form scratch for a given language is expensive and one of the main barriers in porting current systems to many languages is the large amount of data usually needed to train the models of current recognizers. On the other hand, large databases already exist for many languages and acoustic model training may in principle benefit from data in languages other than the target language, assuming that all sounds produced by speakers across languages, share a common acoustic space.

Since we only deal with monolingual language models, the corresponding pronunciation lexicons are monolingual as well. However, in the context of accented speech, dialects or recordings from countries with multiple official languages such as Switzerland, foreign words may appear. Furthermore, a pronunciation lexicon is usually distributed with the database. Typically, acoustic-phonetic model training relies on the data transcription, which is derived from the database-specific lexicon. Even in monolingual environments, lexical resources may differ greatly across databases. This lexical diversity can be very challenging for state-of-the-art ASR systems.

The focus of this thesis is the investigation of multilingual acoustic-phonetic modeling and lexical diversity across languages and databases. Conventional acoustic modeling approaches include HMM/GMM [Rabiner, 1989], where each state is parametrized with a generative Gaussian mixture model (GMM) and hybrid HMM/MLP [Morgan and Bourlard, 1995], where the emission probability of the HMM state is estimated with a discriminative multilayer perceptron (MLP). Most multilingual acoustic modeling found in literature used HMM/GMM based ASR systems [Schultz and Waibel, 2001, Köhler, 2001]. Vu et al. [2011], for example, presented a framework to rapidly build an HMM/GMM system based on multilingual training. We investigated the performance of speech recognition systems with different features and acoustic modeling techniques for multilingual speech recognition [Imseng et al., 2010]. That study on isolated word recognition revealed that multilingual MLP based features and discriminative acoustic modeling techniques, such as MLPs, seem to be well suited for multilingual ASR. Therefore, we study hybrid HMM/MLP based approaches, where the MLP is trained to estimate posterior probabilities (posteriors) of the subword unit that is associated with the HMM state, given the acoustics.

Indeed, posterior based hybrid HMM/MLP systems seem better suited than HMM/GMM systems for such multilingual setups since the MLP can be trained on data from multiple languages[1]. Usually, the MLP is trained to estimate emission probabilities of the HMM states. However, the structure of the HMM used for decoding is monolingual. Hence, the posterior estimates of the MLP may be diverging from the HMM state emission probabilities. Even in monolingual setups, such a mismatch can be introduced, especially if a system is trained across databases that use different lexical resources. Such mismatches can lead to performance degradation and have been addressed in the past through adaptation techniques such as probabilistic acoustic mapping (PAM) [Sim, 2009].

One alternative to avoid mismatch between HMM states and MLP outputs, is to use the posterior estimates of the MLP as features (posterior features) as done for example in Tandem systems [Hermansky et al., 2000]. More specifically, in a Tandem system, the HMM states are modeled with GMMs. However, posteriors are not normally distributed as assumed by the GMMs and, therefore, need to be post-processed. Usually, the logarithm is used to gaussianize the posteriors, followed by a dimensionality reduction transformation. Indeed, posterior

[1] Of course, the GMMs could also be trained on multilingual data, but GMMs 1) are not discriminant and 2) may require too many Gaussians and parameters.

features have successfully been used in multilingual setups [Tòth et al., 2008, Stolcke et al., 2006]. However, in contrast to the hybrid system which directly uses the posteriors as emission probabilities and does not involve any HMM parameter training, the GMMs of the Tandem system need to be trained. If the target language is lacking resources, this may be an intractable problem due to data sparsity.

Recently, Kullback–Leibler divergence based hidden Markov models (KL-HMMs) were introduced [Aradilla, 2008]. KL-HMM is an HMM based system that is able to use raw posterior features and models the states with trained posterior distributions. These reference posteriors are trained by minimizing the Kullback–Leibler (KL) divergence between posterior features and reference posteriors. Such a systems allows the utilization of data from different languages during MLP and HMM training because the relation between the languages can be learned during the reference posterior training which only requires small amounts of data. However, so far, KL-HMM was only investigated in monolingual setups and with context-independent HMM states and MLP outputs, mainly due to the lack of a decision tree algorithm able to handle KL-HMM acoustic models.

1.2 Objective

The goal of this thesis is to investigate posterior based approaches towards the development of multilingual ASR systems and the exploration of language adaptive methods that provide means to build systems for languages lacking resources while focusing on problems related to multilingual acoustic-phonetic modeling and lexical diversity. In this context, we look for principled approaches towards solving acoustic modeling issues related to phonetic mismatches between languages, multilingual features and fast adaptation of systems.

By further exploring multilingual aspects in posterior based ASR, we aim at improving the performance of current monolingual state-of-the-art systems, ideally on high variability recordings of accented speech and dialects. By leveraging similarities across languages, we expect the new system to be more flexible and easy to adapt. Such a system also performs well when having access to a limited amount of data. In the longer term, the research should also lead to ASR systems that are able to deal with unseen languages or languages without written form.

1.3 Main contributions

- Extension of the KL-HMM acoustic modeling approach along two directions:
  - Development of a decision tree clustering algorithm that allows us to build a recognizer based on context-dependent subword units [Imseng et al., 2012d].
  - Integration of high dimensional posterior features estimated by an MLP trained on context-dependent targets [Imseng et al., 2013b].

In such a setup, KL-HMM allows the HMM and the MLP to be trained on different data. The MLP can be trained on large amounts of data in any language and optimally utilize the data by adjusting the number of MLP outputs. A larger number of MLP outputs projects the acoustics into a higher dimensional space, allowing a more subtle distinction of acoustic samples. The HMM, on the other hand, can be trained on low amounts of target language data and still exploit the multilingual information in the form of posterior features. The decision tree clustering allows parameter sharing through state tying and permits the adaptation of the number of HMM states to the amount of available target language data.

- Development of a speaker adaptation method for the KL-HMM framework, referred to as speaker adaptive KL-HMM. Speaker adaptive KL-HMMs express the reference posteriors as a linear regression between reference vectors trained on generic posterior features and reference vectors trained on speaker-specific posterior features [Imseng and Bourlard, 2013].

- Investigation of stochastic phone space transformations across databases and languages to address lexical diversity. The studied soft mapping strategies outperform other mapping strategies including data-driven and knowledge based manual mapping on non-native speech recognition [Imseng et al., 2013a].

- Theoretical and experimental comparisons of the KL-HMM framework with similar approaches such as probabilistic acoustic mapping, supporting the efficiency of the proposed ASR system when dealing with non-native data [Imseng et al., 2013a].

- Exploitation of multiple out-of-language databases to boost the performance of a monolingual under-resourced ASR system. Indeed, in the case of Afrikaans, Dutch, the most similar of the investigated languages yields the best performance [Imseng et al., 2012c].

- Comparison of the KL-HMM acoustic modeling technique to other approaches on an under-resourced monolingual ASR task. In this case, the performance of KL-HMM was compared to posterior feature based approaches such as Tandem, as well as to short-term spectral feature based approaches such as subspace Gaussian mixture models (SGMMs), showing that the KL-HMM framework seems to be preferable if only small amounts of data are available [Imseng et al., 2013b].

As implied above, a large amount of work presented in this thesis has already been published. The extensions applied to the KL-HMM framework were progressively published in [Imseng et al., 2012b,d, Imseng and Bourlard, 2013]. Some of the non-native work appeared in [Imseng et al., 2013a], albeit the contribution of this thesis is more in-depth. Most of the under-resourced language ASR has been published in [Imseng et al., 2013b].

1.4 Structure

This thesis is structured as follows:

- Chapter 2: Background, defines common terms used in this thesis such as phones and phonemes and gives an overview over current state-of-the-art ASR systems. Two different kind of features, namely cepstral features and posterior features estimated with MLPs are reviewed. Furthermore, HMM based as well as template based acoustic modeling techniques are discussed and the employed evaluation metric is presented. Finally, the databases that will be used in this thesis are introduced. The database description convincingly illustrates that different databases have diverse lexical resources using different phoneme sets, pointing at one of the main problems expatiated in this thesis.

- Chapter 3: Stochastic phone space transformations, specifies a new type of stochastic phone space transformation, able to tackle some of the issues related to acoustic modeling, multilingual adaptation of phones and lexical diversity across databases. More specifically, phone variability and phone set mismatch problems between *source* phones and *target* phones are addressed. In that context, we propose a stochastic phone space transformation technique that allows the conversion of source posteriors into target posteriors of any language and phone format. The proposed transformation is validated with non-native speech recognition experiments, also revealing limitations of this approach.

- Chapter 4: KL-HMM, then revisits the recently proposed KL-HMM approach and extends the existing context-independent KL-HMM framework to context-dependent KL divergence based acoustic modeling. Because only small amounts of non-native data are available, we take Greek data as an example to show that the proposed framework is able to reach the performance of a current state-of-the-art HMM/GMM system trained on 10 hours of data and can outperform conventional acoustic modeling techniques if less than one hour of data is available.

- Chapter 5: Non-native ASR, then reports how we apply the extended KL-HMM framework to non-native speech recognition and how we perform extensive theoretical and experimental comparison of KL-HMM to related approaches such as PAM or linear hidden networks (LHNs) and conventional adaptation techniques such as maximum likelihood linear regression (MLLR).

- Chapter 6: Under-resourced ASR, takes Afrikaans as a representative of an under-resourced language and reports how to boost the performance of an under-resourced Afrikaans ASR system by using already available Dutch data. We use three different acoustic modeling techniques, namely KL-HMM, Tandem as well as SGMMs to successfully exploit available multilingual resources. In the case of Tandem and KL-HMM, this is done through posterior features, estimated by an MLP, and in the case of SGMMs,

through parameter sharing. Furthermore, we also compare the three acoustic modeling techniques to conventional adaptation techniques.

- Chapter 7: Speaker adaptive KL-HMM, introduces a speaker adaptation method for the KL-HMM framework. The speaker adaptive KL-HMM performs a simple, adaptive regression between generic and speaker-specific KL-HMM models.

2 Background

This chapter briefly reviews standard feature extraction and acoustic modeling techniques followed by state-of-the-art system descriptions with focus on multilingual as well as accented ASR. Furthermore, the evaluation metric is introduced and, at the end of the chapter, all the databases used in this thesis are described.

2.1 Notation and definitions

Vectors and matrices are denoted by bold symbols and the superscript $^{\mathsf{T}}$ stands for the transpose operator. Subscripts are used to refer to vector indices or indices related to time and superscripts are used to refer to indices related to different (discrete) classes or locations. $P(.)$ refers to the probability of a discrete random variable and $p(.)$ to the probability density function of a continuous random variable.

The terms regularly used in this thesis are defined hereafter:

- **Phoneme**: a phoneme is defined as the smallest sound unit of a language that discriminates between a minimal word pair [Gold and Morgan, 2000, p. 310].
- **Phone**: humans are able to produce a large variety of acoustic sounds which linguists have categorized into segments called phones. Phones are not necessarily the smallest units to describe sounds but they represent a base set that can be used to describe most languages [Gold and Morgan, 2000, p. 310].
- **IPA**: the international phonetic alphabet (IPA) is a notational standard for the phonetic representation of all languages [IPA, 2013].
- **SAMPA**: the speech assessment methods phonetic alphabet (SAMPA) is a machine-readable phonetic alphabet for a large amount of languages [Wells, 2013].
- **Arpabet**: arpabet is a phonetic transcription code for general American English developed by the advanced research projects agency (ARPA) as a part of their speech

understanding project (1971–1976) [ArpaBet, 2013].

2.2 Feature extraction

An acoustic signal contains many different forms of information. For the speech recognition process, a lot of the information contained in the signal is redundant. State-of-the-art speech recognizers therefore first extract relevant information (features) from the speech signal in an efficient, robust manner [Rabiner and Juang, 1993]. The acoustic feature (observation) $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$ is a sequence of T feature vectors $\boldsymbol{x}_t$. This section briefly reviews two kinds of features, cepstral and posterior features.

2.2.1 Cepstral features

Cepstral features are spectral based features that are derived from the cepstrum of a short term signal. The cepstrum is the Fourier transformed log-spectral magnitude of a signal. The two most common cepstral features are mel-frequency cepstrum coefficients (MFCC) [Davis and Mermelstein, 1980] and perceptual linear prediction (PLP) coefficients [Hermansky, 1990]. MFCC and PLP features are very similar and a good comparison is given by Gold and Morgan [2000, ch. 22]. Both methods derive the feature vector from a filter bank designed according to models of the human auditory system. The main difference between the two lies in the nature of spectral smoothing: for MFCCs, cepstral truncation is applied and for PLPs, an autoregressive model is used [Gold and Morgan, 2000]. The autoregressive model often leads to better noise robustness [Openshaw et al., 1993] and speaker independence [Psutka et al., 2001] than the cepstral truncation. Therefore, the experiments described in this thesis make use of mel-frequency PLP (MF-PLP) features [Young et al., 2006], extracted with the hidden Markov model toolkit (HTK) [Young et al., 2006]. MF-PLP features are based on the mel-scale filterbank instead of the Bark-scale as originally proposed by Hermansky [1990].

2.2.2 Posterior features

Posterior features are posterior probability vectors given the acoustics [Aradilla et al., 2009]. An MLP, as shown in Figure 2.1, can discriminatively be trained to estimate such posterior probabilities of q^d, with $d = 1, \cdots, D$, and D being the total number of MLP outputs[1], given cepstral features $\boldsymbol{X}$, such as MF-PLPs as described in Section 2.2.1 [Richard and Lippmann, 1991]. An MLP can be trained to estimate $P(q^d|\{\boldsymbol{x}_{t-a}, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t+a}\})$ where a stands for the temporal context that is considered. For the ease of notation, the input of the MLP is written as $\boldsymbol{X}_t = [\boldsymbol{x}_{t-a}^\mathsf{T}, \cdots, \boldsymbol{x}_t^\mathsf{T}, \cdots, \boldsymbol{x}_{t+a}^\mathsf{T}]^\mathsf{T} = [x_1, \cdots, x_K]^\mathsf{T}$, where $K = (2a+1)C$, is the number of inputs, with C being the dimensionality of the cepstral features.

[1] MLP outputs are uniquely assigned to HMM states in hybrid systems. We therefore use the same notation q^d for an HMM state and an MLP output.

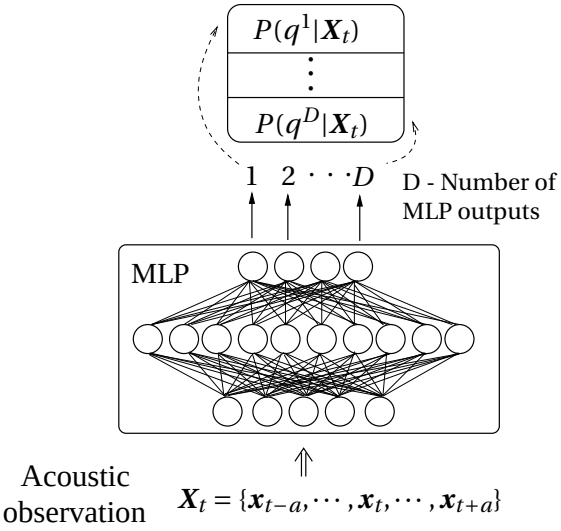


Figure 2.1: A multilayer perceptron taking $\boldsymbol{X}_t = \{\boldsymbol{x}_{t-a}, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t+a}\}$ as input and estimating $P(q^d|\boldsymbol{X}_t)$.

The MLP depicted in Figure 2.1 has three layers, the input layer, the output layer and a hidden layer in between. The number of input units K is given by $\boldsymbol{X}_t$ and the number of output units D (phones) is determined by the language of the training data. Those phones can for example be context-independent monophones [Morgan and Bourlard, 1990] or context-dependent triphones [Bourlard et al., 1992]. The number of hidden units is a parameter.

The output of the neural network, $\boldsymbol{g}(\boldsymbol{X}_t) = [g_1(\boldsymbol{X}_t), \cdots, g_d(\boldsymbol{X}_t), \cdots, g_D(\boldsymbol{X}_t)]^\top$, is a D-dimensional vector, where $g_d(\boldsymbol{X}_t)$ can be written as [Bishop, 2006]:

$$g_d(\boldsymbol{X}_t) = h^o\Big(\sum_{j=0}^{M} w^o_{dj} h^h\Big(\sum_{i=0}^{K} w^h_{ji} x_i\Big)\Big), \tag{2.1}$$

where the variable M stands for the number of hidden units and w^h_{ji} and w^o_{dj} for the weights of the hidden and output layer of the MLP, respectively[2]. The functions h^o and h^h are non-linear functions associated with the output and hidden layer, respectively. Usually, the *sigmoid* function is used as the non-linearity in hidden layers:

$$h^h(y_i) = \frac{1}{1 + exp(-y_i)}, \tag{2.2}$$

with y_i being the weighted sum of inputs. At the output however, it is common to use the *softmax* function to guarantee that the outputs sum to one:

$$h^o(y_j) = \frac{exp(y_j)}{\sum_{\ell=1}^{D} exp(y_\ell)}. \tag{2.3}$$

[2]The bias is absorbed by the weights in (2.1).

MLP training

During training, the weights of the MLP, w^h and w^o, are adjusted using the error back propagation algorithm [Bishop, 2006, ch. 5.3], which requires frame based target values, l_t for every input $\boldsymbol{X}_t$, where l_t stands for the *label* at time t. The algorithm back-propagates the error, measured in terms of a certain cost function such as *mean square error* or *relative entropy*, and then adjusts the weights in the direction of the error gradient with respect to the weights. The relative entropy criterion (sometimes also referred to as Kullback–Leibler distance [Kullback and Leibler, 1951, Kullback, 1987]) can be written as:

$$E = \sum_{d=1}^{D} t_d(\boldsymbol{X}_t) \log \frac{t_d(\boldsymbol{X}_t)}{g_d(\boldsymbol{X}_t)}, \tag{2.4}$$

where $\boldsymbol{t}(\boldsymbol{X}_t) = [t_1(\boldsymbol{X}_t), \cdots, t_d(\boldsymbol{X}_t), \cdots, t_D(\boldsymbol{X}_t)]^\mathsf{T}$ stands for the desired output vector (target), determined from the label l_t and $g(\boldsymbol{X}_t)$ for the observed output vector. The relative entropy criterion is nowadays often referred to as cross-entropy. Relative entropy and cross-entropy are equivalent if binary (hard) targets are used.

To avoid overfitting to the training data, several methods such as early stopping based on cross-validation data have been proposed [Bishop, 2006, ch. 5.5].

In this thesis, we make use of the Quicknet software [Johnson, 2004] to train the employed MLPs on a nine frame temporal context (four preceding and following frames). As we usually do, the number of parameters in the MLPs is set to 10% of the number of available training frames.

MLP forward pass

Once the MLP is trained, the probability $P(q^d|\boldsymbol{X}_t)$ can be estimated with the observed output $g_d(\boldsymbol{X}_t)$. The vector $\boldsymbol{g}(\boldsymbol{X}_t) = [g_1(\boldsymbol{X}_t), \cdots, g_D(\boldsymbol{X}_t)]^\mathsf{T}$ can be used as a feature with [Hermansky et al., 2000] or without [Rigoll and Willett, 1998] processing. Such features are referred to as posterior features.

2.3 Acoustic modeling techniques

Given the acoustic feature vector $\boldsymbol{X}$, the ASR system then aims at decoding $\boldsymbol{X}$ into the most likely sequence of words, $\mathscr{W}^* = \arg\max_{\mathscr{W}} P(\mathscr{W}|\boldsymbol{X})$. Using Bayes' rule, $P(\mathscr{W}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\mathscr{W})P(\mathscr{W})}{p(\boldsymbol{X})}$ and assuming that $p(\boldsymbol{X})$, the average (or prior) probability that $\boldsymbol{X}$ is observed, is constant during decoding, we can formulate the decoding problem as follows:

$$\mathscr{W}^* = \arg\max_{\mathscr{W}} p(\boldsymbol{X}|\mathscr{W})P(\mathscr{W}). \tag{2.5}$$

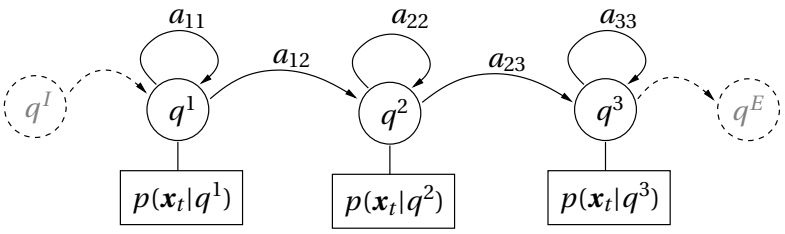


Figure 2.2: A hidden Markov model with three states $\{q^1, q^2, q^3\}$, transition probabilities a_{ij} and probability density distributions associated with the states.

A pronunciation dictionary expands the words into smaller sound units which are modeled by the acoustic model. The acoustic model then estimates the probability that a sequence of acoustic vectors $\boldsymbol{X}$ is observed when a word sequence $\mathscr{W}$ is uttered, $p(\boldsymbol{X}|\mathscr{W})$, and the language model estimates the probability of a word sequence $P(\mathscr{W})$. The decoder finally selects the most likely word sequence by efficiently searching large amounts of possible word sequences.

The acoustic modeling techniques that are briefly reviewed in this chapter can broadly be classified into two categories: HMM based and template based acoustic modeling techniques.

2.3.1 Hidden Markov model based acoustic modeling techniques

One possibility to model the probability $p(\boldsymbol{X}|\mathscr{W})$ is to use an HMM [Rabiner, 1989]. A continuous first order HMM as given in Figure 2.2, is defined by five elements [Rabiner, 1989, Schultz, 2006].

1. Set of emitting states $\{q^1, \cdots, q^D\}$ plus non-emitting initial and end state, q^I and q^E, respectively

2. Continuous observations $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_T\}$

3. State transition probabilities $a_{ij} = P(q_t = q^j | q_{t-1} = q^i)$, where a_{ij} denotes the probability of a transition from state q^i to state q^j with $i, j = 1, \cdots, D$ and q_t being the state at time t

4. Probability density functions to estimate the probability of emitting an observation vector $\boldsymbol{x}_t$ in state q^d at time t, $p(\boldsymbol{x}_t|q_t = q^d)$ (emission probability)

5. Initial state distribution $\pi = \{\pi^1, \cdots, \pi^D\}$ with $\pi^d = P(q_1 = q^d | q^I)$

An HMM comprises two stochastic processes. One stochastic process produces a state sequence $\mathscr{Q} = \{q_1, \cdots, q_t, \cdots, q_T\}$ and the other a sequence of observations according to the probability functions associated with each state. The stochastic process that produces $\mathscr{Q}$ is not directly observable, therefrom *hidden* Markov model.

Given the above HMM definitions, $p(\boldsymbol{X}|\mathscr{W})$ can be rewritten as:

$$p(\boldsymbol{X}|\mathscr{W}) = \sum_{\mathscr{Q}\in\mathbb{Q}^{\mathscr{W}}} p(\boldsymbol{X}|\mathscr{Q})P(\mathscr{Q}), \tag{2.6}$$

where $\mathbb{Q}^{\mathscr{W}}$ denotes the set of all possible state sequences allowed by the word sequence $\mathscr{W}$. Assuming a first order Markov model, i.e. $P(q_t|q_{t-1},\cdots,q_1) = P(q_t|q_{t-1})$ and independent acoustic observations given the state, (2.6) can be rewritten:

$$p(\boldsymbol{X}|\mathscr{W}) \approx \sum_{\mathscr{Q}\in\mathbb{Q}^{\mathscr{W}}} \prod_{t=1}^{T} p(\boldsymbol{x}_t|q_t,\Omega)P(q_t|q_{t-1}), \tag{2.7}$$

where Ω stands for the parameters of the probability density function $p(\boldsymbol{x}_t|q_t,\Omega)$ and $P(q_t|q_{t-1})$ are the transition probabilities $a_{q_t q_{t-1}}$ with $\boldsymbol{A} = (a_{ij})$ being the transition matrix.

HMM training

The transition matrix $\boldsymbol{A}$, and the parameters of the emission probability density function, Ω, form the HMM parameters $\Theta_M = \{\Omega, \boldsymbol{A}\}$, which can be trained using the expectation-maximization (EM) algorithm, a general technique for finding maximum likelihood solutions for probabilistic models having latent variables [Dempster et al., 1977, Gold and Morgan, 2000]. The *full* EM algorithm maximizes the *full* likelihood of the observed data [Gold and Morgan, 2000]:

$$\mathscr{L} = p(\boldsymbol{X}|\mathscr{W},\Theta_M) = \sum_{\mathscr{Q}\in\mathbb{Q}^{\mathscr{W}}} p(\boldsymbol{X},\mathscr{Q}|\Theta_M) = \sum_{\mathscr{Q}\in\mathbb{Q}^{\mathscr{W}}} \prod_{t=1}^{T} p(\boldsymbol{x}_t|q_t,\Omega)a_{q_t q_{t-1}}, \tag{2.8}$$

where $\mathbb{Q}^{\mathscr{W}}$ represents the set of all possible paths in the model of the hypothesized word sequence $\mathscr{W}$. Using the forward procedure, and efficient algorithm to calculate the likelihood $p(\boldsymbol{X}|\mathscr{W},\Theta_M)$, we can rewrite (2.8) as:

$$p(\boldsymbol{X}|\Theta_M) = \sum_{\ell=1}^{D} p(\boldsymbol{X}, q_T^{\ell}|\Theta_M), \tag{2.9}$$

where $\boldsymbol{X}$ is the observed feature sequence of length T, q_T^{ℓ} stands for the event of being in state q^{ℓ} at time T, and D is the number of emitting states in the HMM. Further decomposing (2.9) yields the following, also known as forward recurrence [Gold and Morgan, 2000]:

$$P(q_t^d, \boldsymbol{X}_{1\cdots t}|\Theta_M) = \sum_{\ell=1}^{D} P(q_{t-1}^{\ell}, \boldsymbol{X}_{1\cdots t-1}|\Theta_M)p(\boldsymbol{x}_t|q_t^d,\Omega)a_{q_{t-1}^{\ell} q_t^d}, \tag{2.10}$$

where $\boldsymbol{X}_{1\cdots t}$ stands for the feature sequence $\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_t\}$.

The full likelihood of a model can also be approximated by the likelihood associated with the most likely sequence of states (path). Hence the sum in (2.10) can be replaced with the max

operator. This approximation is referred to as *Viterbi approximation*.

In practice, often the log-likelihood is maximized. This can be achieved through dynamic programming using the following recursion [Gold and Morgan, 2000]:

$$\log P(q_t^d, \boldsymbol{X}_{1\cdots t}|\Theta_M) = \max_{\ell}\left(\log P(q_{t-1}^{\ell}, \boldsymbol{X}_{1\cdots t-1}|\Theta_M) + \log a_{q_{t-1}^{\ell} q_t^d}\right) + \log p(\boldsymbol{x}_t|q_t^d, \Omega). \quad (2.11)$$

HMM decoding

The goal of the decoding is to find the most likely word sequence $\mathscr{W}^*$ given a sequence of acoustic features $\boldsymbol{X}$:

$$\mathscr{W}^* = \arg\max_{\mathscr{W}} P(\mathscr{W}|\boldsymbol{X}). \quad (2.12)$$

Using Bayes' rule, and given that $p(\boldsymbol{X})$ is constant during decoding, we have:

$$\mathscr{W}^* = \arg\max_{\mathscr{W}} \frac{p(\boldsymbol{X}|\mathscr{W})P(\mathscr{W})}{p(\boldsymbol{X})} = \arg\max_{\mathscr{W}} p(\boldsymbol{X}|\mathscr{W})P(\mathscr{W}). \quad (2.13)$$

Hence, the likelihoods of different word sequences $p(\boldsymbol{X}|\mathscr{W})$ need to be estimated. Using the Viterbi approximation:

$$p(\boldsymbol{X}|\mathscr{W}) \approx \max_{\mathbb{Q}^{\mathscr{W}}} \prod_{t=1}^{T} p(\boldsymbol{x}_t|q_t, \Omega) a_{q_{t-1}q_t}, \quad (2.14)$$

and in the log domain:

$$\log p(\boldsymbol{X}|\mathscr{W}) \approx \max_{\mathbb{Q}^{\mathscr{W}}} \sum_{t=1}^{T} \log p(\boldsymbol{x}_t|q_t, \Omega) + \log a_{q_{t-1}q_t}. \quad (2.15)$$

The probability of a word sequence $P(\mathscr{W})$ is usually estimated by a language model [Rabiner and Juang, 1993, p. 435]. State-of-the-art ASR systems usually employ statistical language models that are trained on large text corpora. Statistical language models estimate the probability of the n^{th} word given the $n-1$ previous words. Most systems investigated in this thesis use bi-gram language models ($n = 2$). The output of the recognizer is then the most likely word sequence $\mathscr{W}^*$.

In practice, usually the probability of the language model, $P(\mathscr{W})$, is scaled before it is multiplied with the probability of the acoustic model, $p(\boldsymbol{X}|\mathscr{W})$, and word transitions are usually penalized by adding a fixed value to each token when it transits from the end of one word to the start of the next. The language model scaling factor and the word insertion penalty can have a significant effect on recognition performance and hence, some tuning on development data is well worthwhile [Young et al., 2006].

HMM based systems using cepstral features

We review here the most common HMM systems using cepstral-like features, as introduced in Section 2.2.1, in the context of multilingual ASR.

HMM/GMM An HMM based ASR system that uses a GMM to model the emission probability, is referred to as *HMM/GMM* system [Rabiner, 1989]. A GMM is a probabilistic model that consists of a mixture of Gaussian distributions:

$$p(\boldsymbol{x}_t|\Omega, q^d) = \sum_{n=1}^{N} c_n^d p_n^d(\boldsymbol{x}_t|\Omega_n^d), \tag{2.16}$$

where $p(\boldsymbol{x}_t|\Omega, q^d)$ stands for the likelihood of an acoustic observation given the parameters $\Omega = \{c_n^d, \Omega_n^d\}$. Hence, each state q^d is parametrized with a mixture of N Gaussians. The probability density function of the n^{th} Gaussian distribution, p_n^d, is parametrized with $\Omega_n^d = \{\boldsymbol{\mu}_n^d, \Sigma_n^d\}$, where $\boldsymbol{\mu}_n^d$ is the mean, Σ_n^d the variance, and $p_n^d(\boldsymbol{x}_t|\Omega_n^d) = \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_n^d, \Sigma_n^d)$.

MLLR and MAP HMM/GMM systems are often used for ASR these days, but require a relatively large amount of data to train all the parameters. Adaptation techniques such as MLLR [Gales, 1998] and maximum a posteriori (MAP) adaptation [Gauvain and Lee, 1993] also exist for scenarios with less data available. These adaptation techniques have broadly and successfully been applied to scenarios such as recognizing accented (non-native) speech [Wang et al., 2003] or performing speaker adaptation [Leggetter and Woodland, 1995]. However, MLLR and MAP seem to be confined to adaptations within a language [Byrne et al., 2000].

SCHMM Instead of using N Gaussians per HMM state, Huang and Jack [1989] proposed to use semi-continuous HMMs (SCHMMs) that use a total of S Gaussians. Each state can then be parametrized as:

$$p(\boldsymbol{x}_t|\Omega, q^d) = \sum_{s=1}^{S} c_s^d p_s(\boldsymbol{x}_t|\Omega_s), \tag{2.17}$$

where the probability density function of the s^{th} Gaussian distribution p_s is parametrized with $\Omega_s = \{\boldsymbol{\mu}_s, \Sigma_s\}$ (shared among all the states) and the weights c_s^d are estimated for each state individually. Since the Gaussian parameters (means and variances) are shared among the states, the required training data may be significantly less for SCHMMs than for continuous HMMs. Therefore, acoustic modeling techniques similar to SCHMMs are often used to share data among different languages [Köhler, 2001, Schultz and Waibel, 2001, Niesler, 2007].

ML-tag Schultz and Waibel [2001] for example, proposed to share parameters in multilingual environments. The language-tagged acoustic modeling technique (ML-tag)

parametrizes each state as follows:

$$p(\boldsymbol{x}_t|\Omega, q^d) = \sum_{n=1}^{N} c_n^d \mathcal{N}(\boldsymbol{x}_t|\Omega_n^d), \tag{2.18}$$

where N is the number of Gaussians used to model state q^d. HMM states across different languages share the Gaussian components Ω_n^d if they are represented with the same IPA symbol. The mixture weights c_n^d however, are trained for each HMM state individually:

$$c_n^i \neq c_n^j \quad , \quad \forall i \neq j, \tag{2.19}$$

$$\Omega_n^i = \Omega_n^j \quad , \quad \forall i, j : \text{ipa}(q^i) = \text{ipa}(q^j). \tag{2.20}$$

The *IPA-OVL* approach of Köhler [2001] is similar in spirit. ML-tag as well as IPA-OVL perform slightly worse than language dependent acoustic models [Köhler, 2001, Schultz and Waibel, 2001]. On the other hand such systems can be used to rapidly develop an ASR system for a new language [Schultz and Waibel, 2001] or to build a system for languages for which only low amounts of training data is available [Niesler, 2007].

SGMM A subspace Gaussian mixture model (SGMM) can be described as follows [Povey et al., 2010]:

$$p(\boldsymbol{x}_t|\Omega, q^d) = \sum_{i=1}^{I} c_i^d \mathcal{N}(\boldsymbol{x}_t|\Omega_i^d), \tag{2.21}$$

where $p(\boldsymbol{x}_t|\Omega, q^d)$ stands for the likelihood of an acoustic observation given the parameters $\Omega = \{c_i^d, \Omega_i^d\}$. All the states share the same I Gaussians, similar to SCHMM. The model in each HMM state is then represented by a simple GMM with I Gaussians, mixture weights c_i^d, means $\boldsymbol{\mu}_i^d$, and covariances Σ_i. The latter are shared across all states. The state-specific mixture weights and means are estimated as follows:

$$\boldsymbol{\mu}_i^d = \mathbf{M}_i \boldsymbol{v}^d, \tag{2.22}$$

$$c_i^d = \frac{\exp(\boldsymbol{w}_i \cdot \boldsymbol{v}^d)}{\sum_{\ell=1}^{I} \exp(\boldsymbol{w}_\ell \cdot \boldsymbol{v}^d)}, \tag{2.23}$$

where $\boldsymbol{v}^d \in R^U$ is a state-specific vector of dimensionality U. The dimensionality U is a parameter of the system and often chosen to be similar to the dimensionality of the input features.The globally shared parameters $\boldsymbol{M} = [\boldsymbol{M}_1, \cdots, \boldsymbol{M}_I]^\mathsf{T}$ and $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_I]^\mathsf{T}$ are used to derive the means and mixture weights representing the given HMM state, where $\boldsymbol{w}_i$ is a U dimensional vector and $\boldsymbol{M}_i$ a $C \times U$ dimensional matrix with C being the dimensionality of the (cepstral) features. The parameter sharing of SGMM can also be exploited in a multilingual environment [Burget et al., 2010].

Note that the equations above assume (without loss of generality) one state-specific vector $\boldsymbol{v}^d$ to be assigned to each HMM state. However, each state can also be modeled

with a mixture of sub-states [Povey et al., 2011].

HMM based systems using posterior features

In this section, HMM systems based on posterior features (see Section 2.2.2) are reviewed. State-of-the-art posterior feature based systems often use *deep MLPs* [Dahl et al., 2012] with many layers and trained in a complex way on huge amounts of data. This work employs standard MLPs with one hidden layer, which could of course be replaced with more complex MLPs, possibly yielding better posterior features. The theoretical aspects of the ASR systems, which use the MLP as feature extractor, are the same for standard and deep MLPs.

Hybrid HMM/MLP Morgan and Bourlard [1995] proposed to use MLPs to estimate the emission probability of an HMM-based system. As described in Section 2.2.2, an MLP can be trained to estimate the probability of an HMM state q^d given the acoustic feature vectors $\boldsymbol{X}_t$, $P(q^d|\boldsymbol{X}_t)$. Applying Bayes' rule and assuming that $p(\boldsymbol{x}_t)$ is constant during recognition $p(\boldsymbol{x}_t|q^d)$ can be estimated based on $P(q^d|\boldsymbol{x}_t)$ and $P(q^d)$:

$$p(\boldsymbol{x}_t|q^d) = \frac{P(q^d|\boldsymbol{x}_t)p(\boldsymbol{x}_t)}{P(q^d)} \propto \frac{P(q^d|\boldsymbol{x}_t)}{P(q^d)} \approx \frac{P(q^d|\boldsymbol{X}_t)}{P(q^d)}, \tag{2.24}$$

where $P(q^d)$ is the prior probability of an HMM state and can be estimated on the training data. Note that the MLP estimates the posterior features, $P(q^d|\boldsymbol{x}_t)$, given the temporal context $\boldsymbol{X}_t = \{\boldsymbol{x}_{t-a}, \cdots, x_{t+a}\}$, $P(q^d|\boldsymbol{X}_t)$.

Several studies explored multilingually trained MLPs for hybrid systems [Dupont et al., 2005, Scanzio et al., 2008]. Dupont et al. [2005] reported improvement on accented and non-native speech using a multilingually trained MLP. Similar to the ML-tag and IPA-OVL systems, Scanzio et al. [2008] found that the systems incorporating a multilingual MLP perform slightly worse than the systems using language-dependent MLPs. However, multilingual MLPs are always shown to be beneficial in the case of non-native speech and/or in the case of insufficient training data.

Tandem HMM based systems typically use GMMs to model acoustic features. The hybrid HMM/MLP system on the other hand, uses discriminatively trained posterior features. Hermansky et al. [2000] introduced a Tandem system, which is an HMM/GMM system that uses posterior features instead of cepstral features. However, the discriminatively trained posterior features are not normally distributed. Usually, the logarithm is used to gaussianize the posteriors. Often, the log-posteriors are also orthogonalized using principal component analysis (PCA).

Posterior features have shown to be relatively easily portable across languages [Stolcke et al., 2006, Tòth et al., 2008].

2.3.2 Template based acoustic modeling techniques

Instead of using HMMs to model the probability $p(\boldsymbol{X}|\mathscr{W})$, template matching (TM) can be used. TM is a general classification technique that relies on the principle that a class W_i, such as a word, can be characterized by a set of samples (templates) $\mathscr{Y}(W_i) = \{\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_N\}$ belonging to that class, with $\boldsymbol{Y}_n = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{T_n}\}$ and T_n the length of template $\boldsymbol{Y}_n$ [Aradilla, 2008].

Template training

During training, a number of reference templates N are collected for each class W_i. Rabiner and Juang [1993] list different methods how to collect those N templates:

Casual training is the simplest training procedure, where each token from the training session is used as a reference pattern. However, that approach usually only works for systems trained for a specific speaker and with a small number of different classes (small vocabulary) [Rabiner and Juang, 1993].

Robust training is a sequential training approach in which a consistent pair of tokens is needed for each class. Therefore, each class needs to be spoken until such a consistent pair is obtained. To determine if a pair of tokens is consistent, usually dynamic time warping (DTW) is used to calculate a distortion score between two tokens. If the distortion score is smaller than a threshold, the pair is considered as consistent. The reference template is then computed as a warped average of the two tokens. Usually only one single robust template is stored per class. However such systems are not suitable for speaker independent tasks [Rabiner and Juang, 1993].

Clustering is an alternative to casual and robust training that allows the implementation of a speaker-independent system. During clustering, all the recorded utterances of a class are clustered into N templates. Within each cluster, the utterances should be similar. To determine how similar two utterances are, a similarity measure $\mathscr{F}(\boldsymbol{X}^1, \boldsymbol{X}^2)$ is computed between two utterances $\boldsymbol{X}^1$ and $\boldsymbol{X}^2$. During decoding, the same similarity measure is used (see next section). Many clustering algorithms have been proposed in literature: supervised, semi-supervised and automatic ones. A good overview is given in [Rabiner and Juang, 1993, Section 5.3.3].

The templates that are collected during training, $\boldsymbol{Y}_n = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{T_n}\}$, are then stored in memory.

Template decoding

To decode a test sample $\boldsymbol{X}$, a similarity measure $\mathscr{F}(\boldsymbol{X}, \boldsymbol{Y})$ is computed between $\boldsymbol{X}$ and each template, $\boldsymbol{Y}$, stored in memory during training. The test utterance $\boldsymbol{X}$ is then decided to belong

to the same class as the template with the lowest similarity measure, $\boldsymbol{Y}_j$,

$$W^* = \underset{W_i \in \mathbb{W}}{\arg\min} \min_{\boldsymbol{Y}_j \in \mathscr{Y}(W_i)} \mathscr{F}(\boldsymbol{X}, \boldsymbol{Y}_j), \tag{2.25}$$

where $\mathbb{W}$ is the set of all possible words. The similarity measure $\mathscr{F}(\boldsymbol{X}, \boldsymbol{Y})$ can be calculated using DTW [Aradilla, 2008]:

$$\mathscr{F}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\phi} \sum_{t=1}^{T} d(\boldsymbol{x}_t, \boldsymbol{y}_{\phi(t)}), \tag{2.26}$$

where $d(\boldsymbol{a}, \boldsymbol{b})$ represents the local distance between two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ and T the duration of the test utterance $\boldsymbol{X}$. The function ϕ maps the vectors from the template to the vectors of the test utterance. A distance in the mathematical sense needs to fulfill positive definiteness, symmetry and triangle inequality. However, for template based ASR often a distortion measure, a measure of difference that only meets the positive definiteness, is used [Rabiner and Juang, 1993].

Evidently, (2.26) resembles (2.15), page 15. An extensive comparison between template based ASR and Viterbi approximated HMM based ASR can be found in [Aradilla, 2008, Section 3.3.3].

Template based systems using cepstral features

Many different local distortion measures have been investigated in literature [Rabiner and Juang, 1993, Nocerino et al., 1985]. There are various cepstral based distortion measures such as the weighted or the truncated cepstral distance that are based on the cepstral coefficients of a signal (see Section 2.2.1). Likelihood based measures such as the Itakura-Saito or the likelihood ratio distortion are based on linear predictive coding (LPC) coefficients. For example Nocerino et al. [1985] compared different distortion measures. Rabiner and Juang [1993] also give an extensive overview over many different variants of cepstral and likelihood based distortion measures. Recently, given the large amount of available training data, template based systems gained new attention [De Wachter et al., 2007]. The system presented in [De Wachter et al., 2007] for instance, is based on cepstral-like features and employs the Mahalanobis distance as local distortion measure.

Template based systems using posterior features

Aradilla [2008] used posterior features to perform template based ASR and utilized the KL divergence as local distortion measure. For example Soldo et al. [2011] showed that the ASR performance in template based systems is sensitive to the choice of features and local distances. If posterior features and a KL divergence based distance measure are used, template based systems can perform better than HMM based hybrid HMM/MLP systems [Soldo et al., 2011].

2.4 Evaluation

It is common to measure the complexity of a recognition task with the perplexity of the language model. Therefore, Section 2.4.1 shows how we estimate the perplexity of a bi-gram language model. To evaluate the performance of different acoustic modeling approaches, the word accuracy is calculated as described in Section 2.4.2. To determine if there is a significant difference between the word accuracies measured for two different decoders, we then use the significance test described in Section 2.4.3

2.4.1 Perplexity of a language model

To estimate the perplexity, a language model and a test word sequence are required. The perplexity of a language model is derived from the entropy $H(\mathscr{W})$ of the test sequence. For bi-gram language models, the entropy can be approximated as [Rabiner and Juang, 1993, p. 449]:

$$H(\mathscr{W}) = -\sum P(\mathscr{W}) \log P(\mathscr{W}) \approx -\frac{1}{N} \sum_{n=1}^{N} \log P(W_n|W_{n-1}) = \hat{H}(\mathscr{W}), \tag{2.27}$$

where N is the total number of words in the test sequence $\mathscr{W}$ and W_n the n^{th} word. The perplexity is then obtained as $2^{\hat{H}(\mathscr{W})}$. Lower perplexity language models are usually sought, although it is known that the perplexity is only loosely correlated with the performance (word accuracy) of an ASR system.

2.4.2 Word accuracy

To compute the word accuracy, first, the output of the decoders need to be compared with the original reference transcriptions. In this work, this is done by using the HTK tool *HResults* [Young et al., 2006] that optimally matches the recognized and reference label sequences by performing dynamic programming as described in detail in [Young et al., 2006]. After this matching procedure, the number of substitution errors (E_S), deletion errors (E_D) and insertion errors (E_I) can be calculated. The percent accuracy is then defined as:

$$\text{Percent Accuracy} = \frac{N - E_D - E_S - E_I}{N} \times 100\%, \tag{2.28}$$

where N is the total number of labels in the reference transcription. Usually, the labels are words, hence the word accuracy is measured. However, if there is no appropriate language model for a database, the labels may also be phonemes. In the latter case, the phoneme accuracy is measured.

2.4.3 Significance test

All the significance tests in this thesis employ the bootstrap estimation method [Bisani and Ney, 2004]. The core idea of the bootstrap estimation method is to create replications of a statistic by random sampling from the data set with replacement. To compare two word accuracies obtained on the same data set, it is crucial that the difference in the number of errors of the two systems are calculated on identical bootstrap samples. For all the significance tests in this thesis, a confidence interval of 95% is used.

2.5 Databases

This section gives an overview over the databases used for the experiments. The phoneme sets employed by the different databases are also listed in Appendix A. Each phoneme set includes one phoneme *sil* that is assigned to silence.

2.5.1 SpeechDat(II) – English, French, German, Greek, Italian and Spanish

SpeechDat is a series of speech data collection projects funded by the European Union. The aim of the SpeechDat data collections is to establish speech databases for the development of voice operated teleservices and speech interfaces. The data collections are standardized, high quality resources to perform speech and language research.

SpeechDat(II) is one of the SpeechDat projects and currently consists of recordings from 14 different European countries. Three different types of SpeechDat(II) databases are available: databases recorded over the fixed telephone network, databases recorded over the mobile network and databases designed for speaker verification. This work only considers fixed telephone network databases, recorded at 8 kHz and stored in uncompressed 8 bit A-law format. A complete list of available databases can be found on the SpeechDat(II) homepage[3].

To be representative, the SpeechDat(II) databases in all languages are gender-balanced, dialect-balanced according to the dialect distribution in a language region and age-balanced. The recorded speakers (500 to 5,000 per database) called a toll free number, answered several questions and read sentences. The databases are intended to be used for developing a number of applications such as information services, transaction services and other call processing services and are subdivided into different corpora. *Corpus S*, used in this work, contains 10 phonetically rich sentences per speaker, which are created artificially to be phonetically balanced. More information, including phoneme frequency statistics, is available in the documentations that come with the databases. To build comparable systems, test sets are specified for every database (depending on the size of the database), and standardized test routines are described by Chollet et al. [1998]. Every language has a dictionary that transcribes the pronounced words in the SAMPA [Wells, 2013] phoneme vocabulary. The employed

[3] http://www.speechdat.org/SpeechDat.html

Table 2.1: Overview over the SpeechDat(II) databases used in this thesis. The number of phonemes and the amount of training, development and test data is given for each language.

ID	Language	# of phonemes	Training data	Development data	Test data
EL	Greek	31	13.5 h	1.5 h	6.9 h
EN	British English	45	12.4 h	1.4 h	4.6 h
ES	Spanish	32	11.5 h	1.3 h	4.3 h
IT	Italian	52	11.5 h	1.3 h	4.3 h
SF	Swiss French	42	13.5 h	1.5 h	4.9 h
SZ	Swiss German	59	14.1 h	1.6 h	5.3 h

phoneme sets are also shown in Table A.1, page 102.

For this work, as shown in Table 2.1, the datasets of six languages are used, namely British English, Greek, Italian, Spanish, Swiss French and Swiss German. In Swiss German, there are 2,000 recorded speakers which are split into a training, development and test set according to the standardized procedure, which preserves the gender, dialect and age distributions of the original set [Chollet et al., 1998]. As standardized by SpeechDat, for datasets with a minimum of 2,000 speakers, the test set consists of 500 speakers. The remaining 1,500 speakers are sub-divided into a development set (10%, 150 speakers) and a training set (1,350 speakers). To avoid any bias in terms of available amount of data towards a particular language, the same number of speakers is used in all languages, even if other databases provide data from more than 2,000 different speakers. For this purpose, a subset of 2,000 speakers is chosen from the whole dataset by using the same procedure as for the test set creation and then that subset is further split into training, development and test set. Hence, rather than using the pre-defined test sets, this work uses the publicly available scripts [Chollet et al., 1998] to ensure that the splits can be reproduced. The amount of training, development and test data for each language as well as the number of phonemes used in the dictionaries are summarized in Table 2.1.

2.5.2 HIWIRE – non-native English

HIWIRE [Segura et al., 2007] is a non-native English speech corpus that contains English utterances pronounced by natives of France (31 speakers), Greece (20 speakers), Italy (20 speakers) and Spain (10 speakers).

The utterances contain spoken pilot orders that are inputs for the controller pilot data link communication (CPDLC) which is a mean of communication between the air traffic controller and the flight crew. CPDLC uses a prompt based vocal input [Segura et al., 2007]. The prompts are described by means of a deterministic grammar, and it contains numbers, spoken letters and common names of instruments and orders. The number of different words is 133 and the grammar perplexity is 14.9. The dictionary is in Carnegie Mellon University (CMU) format and makes use of 38 Arpabet [ArpaBet, 2013] phonemes, also given in Table A.1, page 102.

Table 2.2: Overview over the HIWIRE database. The number of speakers, their mother tongue, the number of utterances and the amount of adaptation and test data is given for each non-native accent. Note that one Spanish accented speaker only recorded 99 utterances.

ID	Mother tongue	# of speakers	# of utterances	Adaptation data	Test data
FR	French	31	3,100	50 min	47 min
GR	Greek	20	2,000	45 min	47 min
IT	Italian	20	2,000	37 min	37 min
SP	Spanish	10	999	18 min	17 min
Total	-	81	8,099	149 min	148 min

The database contains two different kinds of speech material: an original set of utterances, recorded in a quiet room using a close-talking microphone which is referred to as the *clean* partition of the database, and a second set of utterances has been obtained by the addition of noise recorded in a real plane cockpit to the clean data [Segura et al., 2007]. This work only uses the clean partition.

A total of 8,099 English utterances have been recorded from the 81 non-native speakers using a sampling frequency of 16 kHz and stored in 16 bits puls code modulation (PCM) Windows waveform audio file format (WAVE). Hence, HIWIRE consists of 100 recordings per speaker, of which the first 50 utterances are commonly defined to serve as adaptation data and the second 50 utterances as test data. An overview over the different non-native accents and the amount of adaptation and test data is given in Table 2.2.

2.5.3 Lwazi – Afrikaans

Lwazi means *knowledge*. The Lwazi project aims to develop a telephone-based speech-driven information system. The project should provide South Africans with an opportunity to access government information and services in any of South Africa's eleven official languages using either landline telephones or mobile telephones, free of charge [Lwazi, 2013]. This work only makes use of the Afrikaans part of the Lwazi corpus.

The Afrikaans data is available from the Lwazi corpus provided by the Meraka Institute, CSIR, South Africa[4] and described by Barnard et al. [2009]. The database consists of 200 speakers, recorded over a telephone channel at 8 kHz and stored as 16 bit WAVE audio, Microsoft PCM format. Each speaker produced approximately 30 utterances, where 16 were randomly selected from a phonetically balanced corpus and the remainder consisted of short words and phrases.

The Afrikaans database comes with a dictionary [Davel and Martirosian, 2009] that defines the phoneme set containing 38 phonemes, shown in Table A.1, page 102. The dictionary used in this work contained 1,585 different words. The HLT group at Meraka provided us with the training and test sets. In total, about three hours of training data and 50 minutes of test data is

[4] http://www.meraka.org.za/hlt

Table 2.3: Overview over the MediaParl database. The number of phonemes and the amount of training, development and test data is given for both languages.

Language	# of Phonemes	Training data	Development data	Test data
French	38	16.1 h	2.2 h	3.2 h
German	57	14.5 h	2.1 h	4.6 h

available (after voice activity detection). Unfortunately, the Lwazi corpus does not come with an Afrikaans language model.

2.5.4 CGN – Dutch

The spoken Dutch corpus, corpus gesproken nederlands (CGN), [Oostdijk, 2000] contains standard Dutch pronounced by more than 4,000 speakers from the Netherlands and Flanders. The database is divided into several subsets and this work only uses *Corpus o* that contains phonetically aligned *read* speech data pronounced by 324 speakers from the Netherlands and 150 speakers from Flanders. Corpus o uses 47 phonemes, given in Table A.1, page 102, and contains 81 h of data after the deletion of silence segments that are longer than one second. It was recorded at 16 kHz and stored as 16 bit WAVE audio, Microsoft PCM format.

2.5.5 MediaParl – French and German

MediaParl is a Swiss accented bilingual database containing recordings in both French and German as they are spoken in Switzerland [Imseng et al., 2012a]. The data were recorded at the Valais Parliament. Valais is a bilingual Swiss state with many local accents and dialects. Therefore, the database contains data with high variability and is suitable to study multilingual, accented and non-native speech recognition as well as language identification and language switch detection. The database is publicly available for download.

The database consists of recordings of Swiss Valaisan parliament debates of the years 2006 and 2009. The parliament debates always take place in the same closed room. Each speaker intervention can last from about 10 seconds up to 15 minutes. Speakers are sitting or standing when talking and their voice is recorded through a distant microphone. The recordings from 2009 are also available as video streams online[5]. All the audio data (2006 and 2009) is available as WAVE audio, Microsoft PCM, 16 bit, mono 16 kHz. The database contains 7,042 annotated sentences (about 20 hours of speech) for the French language and 8,526 sentences (also about 20 hours of speech) for the German language.

The database is partitioned into training, development and test sets as shown in Table 2.3. The test set contains all the speakers, seven in total, which speak in both languages. Hence, it contains all the non-native utterances. 90% of the remaining speakers, 180 randomly chosen

[5] http://www.canal9.ch/television-valaisanne/emissions/grand-conseil.html

ones, form the training set and the other 10%, 17 speakers, the development set. Note that the different sets are slightly unbalanced between the languages because there are considerable differences in the amount of speech per speaker.

The phonemes in the dictionaries, also shown in Table A.1, page 102, are represented using the SAMPA alphabet. Manual creation of a dictionary can be quite time consuming because it requires a language expert to expand each word into its pronunciation. Therefore, Phonetisaurus [Novak et al., 2012], a grapheme-to-phoneme tool that uses existing dictionaries to derive a finite state transducer based mapping of sequences of letters (graphemes) to their acoustic representation (phonemes), was used to bootstrap the dictionaries with publicly available sources.

To bootstrap the German dictionary, Phonolex [Schiel, 2013] was used. 82% of the German MediaParl words were found in Phonolex. All automatically generated dictionary entries were manually verified in accordance to the German SAMPA rules [Caesar, 2012]. The French dictionary was bootstrapped with BDLEX [Perennou, 1986]. 83% of the French MediaParl words were found in BDLEX. Similar to German, Phonetisaurus was trained on BDLEX to generate the missing pronunciations. Again, all dictionary entries generated with Phonetisaurus were manually verified in accordance to the French SAMPA rules.

2.6 Summary

This chapter gave an overview over two different feature extraction methods: cepstral features and posterior features. Furthermore HMM based as well as template based acoustic modeling techniques were reviewed and the evaluation metric was presented. Then, at the end of the chapter, the databases that will be used in this thesis were described. The database description revealed that different databases have diverse lexical resources using different phoneme sets. In the next chapter, phone space transformations are introduced that are able to handle acoustic modeling problems related to different phoneme sets within a language as well as across languages.

3 Stochastic phone space transformations

This chapter describes a new type of stochastic phone space transformation, able to tackle some of the issues related to acoustic modeling and multilingual adaptation of phones, specifically crafted for HMM/MLP ASR systems, and working directly with posterior distributions. More specifically, phone variability and phone set mismatch problems between *source* phones and *target* phones are addressed. In that context, we propose a stochastic phone space transformation technique that automatically and *optimally* converts conditional source phone posterior probabilities, conditioned on the acoustics, into target phone posterior probabilities. The source and target phones can be in any language, including the same language, and phone format.

The proposed technique estimates a stochastic transformation matrix with the help of a DTW procedure that makes use of a KL divergence based local distance measure and can be applied to non-native and accented speech recognition or used to adapt systems to under-resourced languages.

Taking the non-native English HIWIRE data as an example, and in the context of hybrid HMM/MLP recognizers, we report how to successfully perform mono- and multilingual posterior based stochastic phone space transformations. The resulting soft mapping will be shown to be significantly superior to other types of mapping including manual mapping.

3.1 Introduction

First, we define the terms *phoneme* and *phone*, followed by a discussion about the concept of a common acoustic space and the diversity of lexical resources that are distributed with databases. Then, phone mapping and phone space transformations are introduced.

3.1.1 Phoneme

State-of-the-art HMM based ASR systems such as the ones presented in Chapter 2 typically use phonemes as subword units. A phoneme is defined as the smallest sound unit of a language that discriminates between a minimal word pair [Gold and Morgan, 2000, p. 310]. The set of all phonemes that are used to model speech in a given language is referred to as a *phoneme set*. The creation of a phoneme set and a lexicon that transcribes the words of a language into phonemes needs linguistic expertise and resources. The phoneme set is specific to a language in the sense that two languages could share some, but usually not all, phonemes. The phonemes that are shared across multiple languages (language independent) are sometimes referred to as *polyphonemes* and the phonemes that are language dependent as *monophonemes* [Anderson et al., 1994, Schultz, 2006].

3.1.2 Phone and phone set mismatch

Humans are able to produce a large variety of acoustic sounds which linguists have categorized into segments called phones. Phones are not necessarily the smallest units to describe sounds but they represent a base set that can be used to describe most languages [Gold and Morgan, 2000, p. 310]. Hence, statistical ASR systems usually focus on particular acoustic realizations of phonemes, with specific stationarity properties, which we then refer to as phones. As a consequence of this, it is often difficult to define a phone set that is unique to a specific language, and universally used across different ASR systems. Whilst most phonetic representations such as SAMPA [Wells, 2013] and Arpabet [ArpaBet, 2013] can be represented using the international phonetic alphabet (IPA) [IPA, 2013], the underlying phonetic lexicons do not necessarily use the same subset of IPA symbols.

Even in the context of well defined phone sets, training phone models for ASR remains a challenging task given the high pronunciation variability of words within the same language as well as the variability of the acoustic realization of polyphonemes across languages. Furthermore, in the case of accented speech, phone realizations are often borrowed from two different languages. In that sense, we can define a *phone set*, as a language- or accent-specific set that contains the acoustic realizations of a particular language or accent.

3.1.3 Common acoustic space and diversity of lexical resources

By definition, we assume that all phones across speakers, accents and languages, share a common acoustic space $\mathfrak{X}$, the acoustic space that could be covered by the human articulatory system, also shown in Figure 3.1. The assumption of a common acoustic space $\mathfrak{X}$ is reasonable and usually underpins the approaches based on shared training or adaptation of acoustic models from multiple languages or accents [Köhler, 2001, Schultz, 2006, Burget et al., 2010].

However, lexical resources that are distributed along with the databases across multiple languages can differ greatly, depending upon the definition and number of phonemes, as

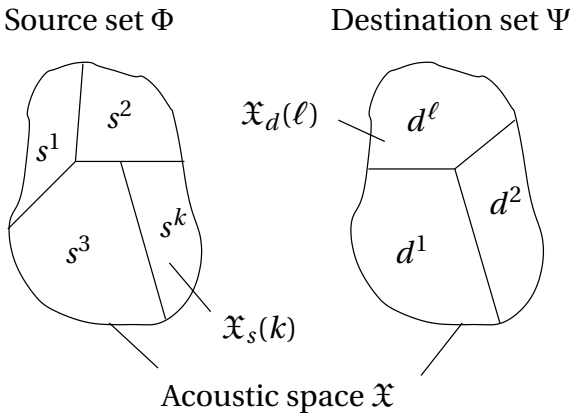


Figure 3.1: Two different sets of phones cover the same acoustic space differently. $\mathfrak{X}_s(k)$ and $\mathfrak{X}_d(\ell)$ are acoustic subspaces associated with phones s^k and d^ℓ respectively.

well as the notation adopted. Furthermore, usually, pronunciation lexicons are created by only taking into account how native speakers pronounce the words. Even then, it is known that acoustic realizations of the same phoneme exhibit high variability, thus, a considerable amount of data is necessary to properly train the models. Modeling variability of the acoustic realizations becomes even more challenging if we have to deal with non-native and accented speech. The main reason is the influence of the native language on the target language sound pronunciation [Van Compernolle, 2001].

3.1.4 Lexical adaptation

One way to handle such problems is to select one notation and have a large lexicon that covers all the possible words. Since spoken language continuously evolves, new words need to be added regularly. Furthermore, to take pronunciation variation into account in the lexicon, often more than one pronunciation per word is needed. However, a general problem of such lexical adaptation approaches is that adding variants to the dictionary can increase confusability between words which can potentially lead to an increase in word error rate [Goronzy et al., 2004].

3.1.5 Phone mapping

An alternate solution is to perform a one-to-one mapping between the phone symbols to adapt the models [e.g Byrne et al., 2000] or share data [e.g. Schultz, 2006]. Such mappings are not limited to context-independent phones, but can also be applied to context-dependent phones such as triphones [e.g Imperl et al., 2000]. Usually, these mappings are manually defined (knowledge based), derived in a data-driven way, or a combination of both. A nice overview over different approaches is given by Schultz [2006, Section 4.4.1].

A phone mapping involves two different phone sets such as the ones given in Figure 3.1.

- A source set consisting of S phones $s^k, k = 1, \cdots, S$.
- A target (destination) set consisting of D phones $d^\ell, \ell = 1, \cdots, D$.

Sim and Li [2008] for example proposed explicit one-to-one probabilistic phone mapping (PPM) that makes use of explicit phonetic reference transcriptions in the form of target phones and outputs of a phone recognizer that uses source phones. As a result, PPM maps each target phone to the most similar source phone. However, a one-to-one mapping between different phone sets may not always exist and even if such a mapping exists, it could be detrimental to the system [Sim, 2009]. The reason for this is partly related to acoustic modeling.

As shown in Figure 3.1, we suppose that there exists an acoustic space that contains all acoustic observations that are involved in the human speech production process. During acoustic modeling, a specific set of phones implicitly partitions this acoustic space into subspaces, each associated with a particular phone. It is possible that two different phone sets can partition the same acoustic space differently, which will not be taken into account during one-to-one mappings.

3.1.6 Phone space transformation

An alternative to one-to-one mappings is phone space transformations. Conceptually, a phone space transformation transforms the subspace that is associated to a source phone during acoustic modeling into a subspace associated to a target phone. Therefore, depending on the acoustic modeling technique, we can distinguish between posterior based phone space transformations and likelihood based phone space transformations.

Posterior based phone space transformations can be applied in the framework of hybrid HMM/MLP systems as follows:

$$P(d_t^\ell|\boldsymbol{x}_t) = \sum_{k=1}^{S} P(d_t^\ell|s_t^k, \boldsymbol{x}_t) P(s_t^k|\boldsymbol{x}_t), \tag{3.1}$$

where $\sum_\ell P(d_t^\ell|s_t^k, \boldsymbol{x}_t) = 1$. Rottland and Rigoll [2000] presented the tied posteriors approach, which considers the special case where the S source phones are context-independent monophones and the target phones are context-dependent triphones, both from the same language. In the present work, we focus on stochastic transformations in general, especially across languages. Furthermore, as we will describe later, we estimate the stochastic transformation matrix differently by directly using phone posteriors instead of converting them to likelihoods and applying the maximum likelihood adaptation as it was done by Rottland and Rigoll [2000].

Sim [2009] extended PPM to probabilistic acoustic mapping (PAM) for hybrid HMM/MLP systems that allows implicit transformation of source posteriors into target

posteriors. PAM significantly outperforms PPM. Indeed our work is similar in spirit to PAM and a detailed comparison between our work and PAM is given later in Chapter 5.

Likelihood based phone space transformation is the corresponding approach for phone space transformations in HMM/GMM systems, where:

$$p(\boldsymbol{x}_t|d_t^\ell) = \sum_{k=1}^{S} c_k^\ell p(\boldsymbol{x}_t|s_t^k), \tag{3.2}$$

with $\sum_k c_k^\ell = 1$. Note that $P(d_t^\ell|s_t^k,\boldsymbol{x}_t)$ are probabilities, hence the sum to one constraint is theoretically founded, whereas c_k^ℓ are weights and the sum to one constraint is arbitrary. Obviously, one-to-one mappings are a particular case of phone space transformations with $P(d_t^\ell|s_t^k,\boldsymbol{x}_t) = \{0,1\}$ or $c_k^\ell = \{0,1\}$, respectively.

SCHMM, as described in Chapter 2, page 16, is a particular kind of likelihood based phone space transformation. Schultz and Waibel [2001] proposed the HMM-based ML-tag method, also summarized in Chapter 2, page 16, to estimate language-independent acoustic models. The approach involves a transformation in the sense that each (multilingual) IPA based universal phone has a pool of S Gaussians. The universal phone model is then transformed to a language specific model by estimating language dependent weights. Our work focuses on hybrid HMM/MLP systems, and not on HMM/GMM systems, but we will show later in Chapter 5 how our work is related to conventional Gaussian mixture based SCHMM systems.

3.2 Posterior based stochastic phone space transformation

In the context of hybrid HMM/MLP recognizers, a stochastic phone space transformation can be formulated as follows. Given an MLP of parameters $\Theta_{\mathscr{S}}$, trained to estimate source phone posterior probabilities, conditioned on acoustic observations, to perform hybrid decoding, we aim to use the already trained MLP to perform ASR on a different database that makes use of a target phone set. Therefore, the source phone posterior estimates need to be transformed to target phone posteriors. Of course, the *source* MLP $\Theta_{\mathscr{S}}$ can also be trained on a mixture of languages to make it more amenable to cross-language adaptation/training.

3.2.1 Model

Transforming source phone posteriors into target phone posteriors then requires the training of a stochastic matrix of parameters:

$$\Theta_M = \begin{bmatrix} P(d^1|s^1)\cdots P(d^D|s^1) \\ \vdots \quad \ddots \quad \vdots \\ P(d^1|s^S)\cdots P(d^D|s^S) \end{bmatrix}, \tag{3.3}$$

where $P(d^\ell|s^k)$ is the probability of a target phone d^ℓ, given a source phone s^k. The matrix Θ_M has dimensionality $S \times D$, where S and D are the number of source and target phones, respectively. The matrix Θ_M will together with the fixed $\Theta_{\mathscr{S}}$ parameterize target phone posterior distributions used as emission probabilities in the HMM/MLP recognizer as follows:

$$P(d_t^\ell|\boldsymbol{X}_t,\Theta) = \sum_{k=1}^{S} P(d_t^\ell|s_t^k,\boldsymbol{X}_t,\Theta)P(s_t^k|\boldsymbol{X}_t,\Theta), \tag{3.4}$$

where $\Theta = \{\Theta_{\mathscr{S}},\Theta_M\}$. The source MLP posteriors, $P(s_t^k|\boldsymbol{X}_t,\Theta)$, are simply estimated by presenting $\boldsymbol{x}_t$, together with some temporal context, at the input of the MLP $\Theta_{\mathscr{S}}$. The conditional target posterior, $P(d_t^\ell|s_t^k,\boldsymbol{X}_t,\Theta)$, is conditioned on the current input $\boldsymbol{X}_t$ and the source phone s^k pronounced at time t.

We assume the following:

- The conditional probability, $P(d_t^\ell|s_t^k,\boldsymbol{X}_t,\Theta)$, can be seen as a similarity measure between a source phone s^k and a target phone d^ℓ. It can thus be assumed time invariant and independent of the acoustic observation $\boldsymbol{X}_t$.

- The source phone posteriors, $P(s_t^k|\boldsymbol{X}_t,\Theta)$, are obtained with the MLP that was previously trained on an independent, frame-level labeled, database that may contain speech of the same language, a different language, or from multiple languages. Since frame-level labeling is available for the source database, the source phone posterior probability estimates are considered independent of Θ_M.

Hence, we can rewrite (3.4) as:

$$P(d_t^\ell|\boldsymbol{X}_t,\Theta) = \sum_{k=1}^{S} P(d^\ell|s^k,\Theta_M)P(s_t^k|\boldsymbol{X}_t,\Theta_{\mathscr{S}}). \tag{3.5}$$

During the training of Θ_M, we assume to have access to a limited amount of target language training data $\boldsymbol{X} = \{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_T\}$ only, which is not labeled in terms of source phones but only in terms of target phones. Furthermore, we assume that no target phone segmentation is available. More specifically, we can associate a target phone class sequence $\{d_1^i,\cdots,d_N^j\}$ with $\boldsymbol{X}$, where $i,j \in \{1,\cdots,D\}$ and N is the number of phones needed to transcribe $\boldsymbol{X}$. The proposed approach will exploit DTW during training (Section 3.2.2), where reference posteriors are associated with target phones. During recognition (Section 3.2.3), the target phone class posterior estimates, $P(d_t^\ell|\boldsymbol{X}_t,\Theta)$, can then be used to perform ASR with a standard hybrid HMM/MLP system on the target database.

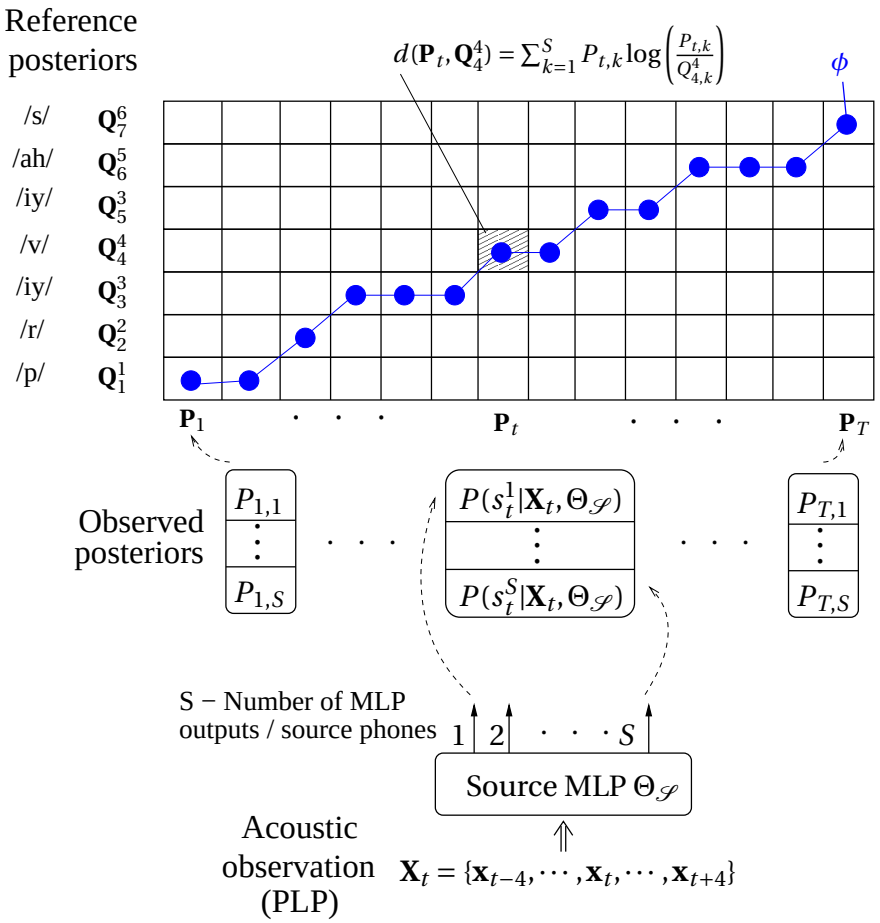


Figure 3.2: Illustration of the DTW based training of the reference posteriors $\boldsymbol{Q}_{\phi(t)}^{\ell}$.

3.2.2 Training

Since the training data $\boldsymbol{X}$ is only transcribed in terms of target phones, we can only estimate $P(s^k|d^\ell, \Theta_M)$ from the source posteriors $P(s_t^k|\boldsymbol{X}_t, \Theta_{\mathscr{S}})$. Applying Bayes' rule to $P(d^\ell|s^k, \Theta_M)$ in (3.5) yields:

$$P(d_t^\ell|\boldsymbol{X}_t, \Theta) = \sum_{k=1}^{S} \frac{P(s^k|d^\ell, \Theta_M) P(d^\ell|\Theta_M)}{\sum_{j=1}^{D} P(s^k|d^j, \Theta_M) P(d^j|\Theta_M)} P(s_t^k|\boldsymbol{X}_t, \Theta_{\mathscr{S}}). \tag{3.6}$$

Given $P(s_t^k|\boldsymbol{X}_t, \Theta_{\mathscr{S}})$, the estimation of $P(d_t^\ell|\boldsymbol{X}_t, \Theta)$ thus requires us to estimate the conditional probability $P(s^k|d^\ell, \Theta_M)$ and the prior probability $P(d^\ell|\Theta_M)$.

Estimation of the conditional probability $P(s^k|d^\ell, \Theta_M)$

The estimation of $P(s^k|d^\ell, \Theta_M)$ is performed through an iterative Viterbi-like based segmentation–optimization training procedure. As illustrated in Figure 3.2, this requires that we first forward pass all the training data $\boldsymbol{X}$ through the source MLP $\Theta_{\mathscr{S}}$ to obtain

$\mathscr{P} = \{\boldsymbol{P}_1, \cdots, \boldsymbol{P}_t, \cdots, \boldsymbol{P}_T\}$, with $\boldsymbol{P}_t$:

$$\boldsymbol{P}_t = \boldsymbol{P}(\boldsymbol{s}|\boldsymbol{X}_t, \Theta_{\mathscr{S}}) = \begin{bmatrix} P(s_t^1|\boldsymbol{X}_t, \Theta_{\mathscr{S}}) \\ \vdots \\ P(s_t^S|\boldsymbol{X}_t, \Theta_{\mathscr{S}}) \end{bmatrix} = \begin{bmatrix} P_{t,1} \\ \vdots \\ P_{t,S} \end{bmatrix}. \tag{3.7}$$

We then use $\boldsymbol{P}_t$, with $t = 1, \cdots, T$, as observed feature vectors alongside with the target phone transcriptions, to train reference posteriors. As shown in Figure 3.2, given the phone transcription of $\boldsymbol{X}$ (i.e. /p/ /r/ /iy/ /v/ /iy/ /ah/ /s/), we can build a sequence of reference posteriors $\mathscr{Q} = \{\boldsymbol{Q}_1^i, \cdots, \boldsymbol{Q}_n^\ell, \cdots, \boldsymbol{Q}_N^j\}$, where $\boldsymbol{Q}^\ell$ is the reference posterior associated with target phone d^ℓ, with $i, j, \ell \in \{1, \cdots, D\}$, D being the number of target phones and N the number of phones needed to transcribe $\boldsymbol{X}$ in terms of target phones.

The reference posteriors $\boldsymbol{Q}^\ell$:

$$\boldsymbol{Q}^\ell = \boldsymbol{P}(\boldsymbol{s}|d^\ell, \Theta_M) = \begin{bmatrix} P(s^1|d^\ell, \Theta_M) \\ \vdots \\ P(s^S|d^\ell, \Theta_M) \end{bmatrix} = \begin{bmatrix} Q_1^\ell \\ \vdots \\ Q_S^\ell \end{bmatrix}, \tag{3.8}$$

are trained based on source posteriors $\boldsymbol{P}_t$. Hence, the dimensionality of $\boldsymbol{Q}^\ell$ is S, the total number of source classes. The global distortion between the observed posterior sequence $\mathscr{P}$ and the reference posterior sequence $\mathscr{Q}$ can then be written as:

$$\mathscr{F}(\mathscr{P}, \mathscr{Q}) = \min_{\{\phi\}^{\mathscr{W}}} \sum_{t=1}^{T} d(\boldsymbol{P}_t, \boldsymbol{Q}_{\phi(t)}^\ell), \tag{3.9}$$

where $\{\phi\}^{\mathscr{W}}$ stands for all the possible paths allowed by the hypothesized word sequence, such as *previous* in the case of Figure 3.2. A path ϕ through the distance matrix, maps the observed posteriors to the reference posteriors with

$$\phi(t) = n \in \{1, \cdots, N\}, \tag{3.10}$$

$$\phi(1) = 1, \tag{3.11}$$

$$\phi(T) = N, \tag{3.12}$$

$$\phi(t+1) = \begin{cases} \phi(t) \\ \phi(t) + 1, \end{cases} \tag{3.13}$$

and $d(\boldsymbol{P}_t, \boldsymbol{Q}_{\phi(t)}^\ell)$ is the local distance measure.

From template based ASR experiments [Soldo et al., 2011], we know that a local distance measure based on the KL divergence [Kullback and Leibler, 1951], sometimes also referred to as relative entropy, between the observed feature vectors $\boldsymbol{P}_t$ and the reference vectors $\boldsymbol{Q}^\ell$ is

appropriate because $\boldsymbol{P}_t$ and $\boldsymbol{Q}^\ell$ are both posterior probability distributions[1]:

$$d(\boldsymbol{P}_t, \boldsymbol{Q}^\ell_{\phi(t)}) = \sum_{k=1}^{S} P(s_t^k|\boldsymbol{x}_t, \Theta_{\mathscr{S}}) \log\left(\frac{P(s_t^k|\boldsymbol{x}_t, \Theta_{\mathscr{S}})}{P(s^k|d_n^\ell, \Theta_M)}\right) = \sum_{k=1}^{S} P_{t,k} \log\left(\frac{P_{t,k}}{Q^\ell_{\phi(t),k}}\right). \tag{3.14}$$

As illustrated in Algorithm 1, the training of the reference posteriors consists of iteratively minimizing the global distortion $\mathscr{F}(\mathscr{P},\mathscr{Q})$ in (3.9) in the Q^ℓ space (optimization step) and ϕ space (segmentation step) respectively. The Kullback–Leibler divergence is convex [Cover and Thomas, 1991, Theorem 2.7.2, p. 30]. Therefore, convergence can easily be proved since at every segmentation and re-estimation step the same global distortion is minimized, respectively in the Q^ℓ and ϕ space. We run the algorithm until convergence.

Algorithm 1 Training of reference posteriors

Step 0: Initialization of Q_k^ℓ
for all $\ell \in \{1, \cdots, D\}$ and $k \in \{1, \cdots, S\}$ **do**

$$Q_k^\ell = \begin{cases} \frac{1}{S}, & \text{if } d^\ell \notin \text{source set } \Phi \\ 1-(S-1)\varepsilon, & \text{if } d^\ell \in \Phi \text{ and } s^k = d^\ell \\ \varepsilon, & \text{if } d^\ell \in \Phi \text{ but } s^k \neq d^\ell \end{cases}$$

ε being small, but positive.
end for
Step 1: Segmentation:
Given $\boldsymbol{P}_t \ \forall\ t$ and $\boldsymbol{Q}^\ell \ \forall\ \ell$, minimize the global distortion $\mathscr{F}(\mathscr{P},\mathscr{Q})$ in (3.9) to find the best mapping ϕ of observed posteriors $\boldsymbol{P}_t$ to reference posteriors $\boldsymbol{Q}^\ell$, i.e. the best path.
Step 2: Optimization:
for all $\ell \in \{1, \cdots, D\}$ **do**
  for all n such that $\boldsymbol{Q}_n^\ell$ exists in $\mathscr{Q}$ **do**
    Find all $\boldsymbol{P}_t$ such that $\phi(t) = n$ and use (3.21) to re-estimate Q^ℓ.
  end for
end for
Iterate step 1 and 2 until convergence.

To minimize $\mathscr{F}(\mathscr{P},\mathscr{Q})$ subject to the constraint that $\sum_{k=1}^{S} Q_k^\ell = 1$, we introduce the Lagrange multiplier λ, take the partial derivative with respect to each variable Q_k^ℓ and set it to zero:

$$\frac{\partial}{\partial Q_k^\ell}\left(\mathscr{F}(\mathscr{P},\mathscr{Q}) + \lambda\left(\sum_{k=1}^{S} Q_k^\ell - 1\right)\right) = \frac{\partial}{\partial Q_k^\ell}\left(\min_{\phi} \sum_{t=1}^{T} d(\boldsymbol{P}_t, \boldsymbol{Q}^\ell_{\phi(t)}) + \lambda\left(\sum_{k=1}^{S} Q_k^\ell - 1\right)\right) = 0. \tag{3.15}$$

[1] Note that the Kullback–Leibler divergence has no upper bound, which, during decoding, may theoretically result in different dynamic ranges for the local scores of different HMM state distributions. Similar measures that have an upper bound, such as the Jensen–Shannon divergence, exist. However, in the case of the Jensen–Shannon divergence, the derivation of the update function, presented in this section, does not have a closed form solution.

Solving (3.15) while keeping ϕ fixed yields:

$$\frac{\partial}{\partial Q_k^\ell}\left(\sum_{n:\boldsymbol{Q}_n^\ell \text{ in } \mathcal{Q}}\sum_{t:\phi(t)=n}\sum_{k=1}^{S} P_{t,k}\log\left(\frac{P_{t,k}}{Q^\ell_{\phi(t),k}}\right)\right)+\lambda = 0, \tag{3.16}$$

$$-\sum_{n:\boldsymbol{Q}_n^\ell \text{ in } \mathcal{Q}}\sum_{t:\phi(t)=n}\frac{P_{t,k}}{Q^\ell_{\phi(t),k}}+\lambda = 0, \tag{3.17}$$

$$\frac{1}{\lambda}\sum_{n:\boldsymbol{Q}_n^\ell \text{ in } \mathcal{Q}}\sum_{t:\phi(t)=n} P_{t,k} = Q_k^\ell. \tag{3.18}$$

The sum to one constraint $\sum_{k=1}^{S} Q_k^\ell = 1$ guarantees:

$$\sum_{k=1}^{S} Q_k^\ell = \sum_{k=1}^{S}\frac{1}{\lambda}\sum_{n:\boldsymbol{Q}_n^\ell \text{ in } \mathcal{Q}}\sum_{t:\phi(t)=n} P_{t,k} = 1. \tag{3.19}$$

Solving (3.19) for λ yields:

$$\lambda = \sum_{n:\boldsymbol{Q}_n^\ell \text{ in } \mathcal{Q}}\sum_{t:\phi(t)=n}\sum_{k=1}^{S} P_{t,k} = \sum_{n:\boldsymbol{Q}_n^\ell \text{ in } \mathcal{Q}}\sum_{t:\phi(t)=n} 1 = T^\ell, \tag{3.20}$$

where T^ℓ is the number of observed posteriors that are associated with a reference posterior $\boldsymbol{Q}^\ell$. Hence, combining (3.18) and (3.20), Q_k^ℓ can be estimated as:

$$Q_k^\ell = \frac{1}{T^\ell}\sum_{n:\boldsymbol{Q}_n^\ell \text{ in } \mathcal{Q}}\sum_{t:\phi(t)=n} P_{t,k}, \tag{3.21}$$

which is nothing else but the arithmetic mean of all the observed posteriors associated with the reference posterior $\boldsymbol{Q}^\ell$.

For initialization, we may make use of prior knowledge as described in Algorithm 1. If the IPA symbol of the target phone d^ℓ is not present in the source phone set, $\boldsymbol{Q}^\ell$ is initialized uniformly. If the IPA symbol of d^ℓ and s^k are same however, all the components of $\boldsymbol{Q}^\ell$ are set to a small positive value ε except for the corresponding component $\boldsymbol{Q}_k^\ell$ which is set to $1-(S-1)\varepsilon$. Since the local distance measure involves the computation of the KL divergence between $\boldsymbol{P}_t$ and $\boldsymbol{Q}^\ell$, given in (3.14), we need to ensure that $\boldsymbol{Q}^\ell$ does not contain zeros. Experiments have shown that uniform initialization will usually yield similar results, although with slower convergence.

Estimation of the prior probability $P(d^\ell|\Theta_M)$

Prior probabilities $P(d^\ell|\Theta_M)$ can be estimated as the relative count of number of observed posteriors associated with a reference posterior, i.e.:

$$P(d^\ell|\Theta_M) = \frac{T^\ell}{T}, \tag{3.22}$$

where T^ℓ is the number of observed posteriors that are associated with a reference posterior $\boldsymbol{Q}^\ell$ and T is the total number of observed posteriors.

3.2.3 Recognition

Given an acoustic test sequence $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$, we first use the source MLP $\Theta_{\mathscr{S}}$ to estimate the source phone posteriors $\boldsymbol{P}_t$. To perform standard hybrid HMM/MLP decoding (see Chapter 2, page 18), the emission probabilities $P(d_t^\ell|\boldsymbol{x}_t,\Theta)$, are then estimated according to (3.6), page 33, by performing a phone space transformation with $P(s^k|d^\ell,\Theta_M) = Q_k^\ell$. After division by the priors $P(d^\ell|\Theta_M)$, we obtain scaled likelihoods $p(\boldsymbol{x}_t|d^\ell,\Theta)$:

$$p(\boldsymbol{x}_t|d^\ell,\Theta) = \sum_{k=1}^{S} \frac{Q_k^\ell}{\sum_{j=1}^{D} Q_k^j P(d^j|\Theta_M)} P_{t,k}, \tag{3.23}$$

which are used as local scores during Viterbi decoding, as presented in (2.15), page 15 and (2.24), page 18.

3.3 Validation experiments on non-native ASR

We study the proposed approach by applying it to non-native speech recognition. We start with the hypothesis that the stochastic phone space transformation is beneficial for non-native and accented speech because we can train the source MLP, $\Theta_{\mathscr{S}}$, with large amounts of multilingual data and then handle the variability in pronunciations with relatively small amounts of data by learning the transformation parameters Θ_M.

For the initial experiments, we first estimate English phone posteriors on SpeechDat(II) data. The non-native target database (HIWIRE) uses a different phonetic lexicon, thus the estimated English phone posteriors need to be transformed. Subsequently, we also estimate universal phone posteriors that are trained on the data of five European languages. We expect the multilingually trained source MLP to yield improvement compared to the monolingually trained MLP.

Furthermore, we also compare the proposed posterior transformations to manual as well as data-driven phone mappings and to a system directly trained on the target database.

3.3.1 Monolingual posterior transformation

To study monolingual posterior based stochastic phone space transformations, we use the HIWIRE and the British English SpeechDat(II) databases (see also Section 2.5, page 22).

HIWIRE is a non-native English speech corpus that contains English utterances pronounced by natives of France (31 speakers), Greece (20 speakers), Italy (20 speakers) and Spain (10

Table 3.1: Summary of the MLP training on SpeechDat(II) British English and multilingual data. The total amount of training data, the frame accuracy on the development data, as well as the source set including the number of phones (S) is given.

System	Source Phone set	# phones	TRN data	DEV frame accuracy
MLP-EN	SAMPA English	45	12.4 h	58.8%
MLP-sUNI	SAMPA universal phone set	117	12.7 h	52.0%
MLP-UNI	SAMPA universal phone set	117	63.0 h	57.5%

speakers). The utterances contain spoken pilot orders made up of 133 words and the database also provides a grammar with a perplexity of 14.9 [Segura et al., 2007]. The dictionary is in CMU format and makes use of 38 Arpabet phonemes (see Table A.1, page 102). HIWIRE consists of 100 recordings per speaker, of which the first 50 utterances are commonly defined to serve as adaptation data and the second 50 utterances as testing data. A detailed overview with the amount of adaptation and test data for each non-native accent is given in Table 2.2, page 24.

The British English SpeechDat(II) database contains native speech and is gender-balanced, dialect-balanced according to the dialect distribution in United Kingdom and age-balanced. The database was recorded over the telephone at 8 kHz and is subdivided into different corpora. We only use *Corpus S*, that contains ten read sentences per speaker. The dictionary that comes with the databases uses 45 phonemes in SAMPA format (see Table A.1, page 102).

The two databases come with different dictionaries using diverse phoneme sets. In this section, we will therefore use the adaptation data of the HIWIRE corpus to transform source posteriors estimated with an MLP, trained on British English SpeechDat(II) data, to target posteriors used to decode HIWIRE data.

Observed Posteriors

To estimate the source phone posteriors $P(s_t^k|\boldsymbol{x}_t,\Theta_{\mathscr{S}})$, we train an MLP, MLP-EN on the SpeechDat(II) data. For the MLP training, we split the databases into training (1500 speakers), development (150 speakers) and testing (350 speakers) sets, according to the procedure described in Section 2.5, page 23. The MLP is then trained from 39 mel-frequency PLP (MF-PLP) features (C_0–$C_{12}+\Delta+\Delta\Delta$) in a nine frame temporal context as input. As usual, the number of parameters in the MLP is set to 10% of the number of available training frames. A summary of the MLP training is given in Table 3.1.

Since HIWIRE was recorded at 16 kHz, the recordings are downsampled to 8 kHz to match the recording conditions of the SpeechDat(II) British English data. Then, the same MF-PLP feature analysis is applied and the data is passed through MLP-EN to estimate the sequence of observed posteriors, $\mathscr{P}$.

System	MLP	WACC
Posteriors trained on native English (no adaptation)	-	90.5 %
Monolingual posterior transformation	MLP-EN (12.4 h)	93.3 %
Multilingual posterior transformation	MLP-sUNI (12.7 h)	94.3 %
Multilingual posterior transformation	MLP-UNI (63.0 h)	96.0 %

Table 3.2: Comparison of monolingual and multilingual posterior transformations on English non-native data. As an additional reference point, but coming from a different implementation [Gemello et al., 2007], the word accuracy (WACC) obtained using posteriors trained on native English is also shown.

Reference Posteriors

To perform ASR on the HIWIRE test set, we estimate $P(s^k|d^\ell,\Theta_M)$ on the adaptation data with the iterative segmentation–optimization procedure, presented in Section 3.2.

Recognition

We use the restrictive grammar rules provided by the HIWIRE database as word level lattices and tune the word insertion penalty on the adaptation data. Standard hybrid decoding is performed using the estimated target phone posteriors $P(d_t^\ell|\boldsymbol{x}_t,\Theta)$. Results of the monolingual posterior based stochastic phone space transformation are given in Table 3.2. As a reference point, Table 3.2 also shows the performance of a hybrid system that uses unadapted posteriors estimated by an MLP trained on native English data (TIMIT, WSJ0-1 and vehic1us-ch0), but coming from a different implementation [Gemello et al., 2007]. Result suggest that the monolingual posterior transformation successfully exploits the adaptation data.

3.3.2 Multilingual posterior transformation

To study multilingual posterior based stochastic phone space transformations, we use the HIWIRE [Segura et al., 2007] database and five SpeechDat(II) databases, namely British English, Italian, Spanish, Swiss French and Swiss German (see also Section 2.5, page 22). Since all the SpeechDat(II) dictionaries use SAMPA symbols, we merge phones that share the same SAMPA symbol across languages to build a *universal phone set*. This knowledge-based approach is often used in literature and usually outperforms data-driven mappings [Grézl et al., 2011].

Similar to the monolingual posterior based stochastic phone space transformation presented in Section 3.3.1, we train an MLP to estimate the observed posteriors. In contrast to the monolingually trained MLP-EN in Section 3.3.1, a universal MLP (MLP-UNI) is trained on all the data from the five SpeechDat(II) databases. Additionally, we also train a small universal MLP (MLP-sUNI) that only uses one fifth of the data, randomly chosen, to match the amount of training data available to MLP-EN. A summary of the MLP training is also given in

Table 3.1. Then, observed and reference posteriors are obtained as discussed in Section 3.3.1 and recognition is performed through target phone posterior based hybrid decoding.

MLP-EN and MLP-sUNI are trained on similar amounts of data. However, we expect MLP-sUNI to perform better than MLP-EN because it is trained on data from multiple languages. Furthermore, we hypothesize that MLP-UNI performs better than MLP-sUNI and MLP-EN because it is trained on larger amounts of multilingual data. Indeed, Table 3.2 confirms both hypotheses and shows that the proposed approach can be used to transform robust universal phone posteriors to monolingual phone posteriors and improve ASR performance on non-native speech.

3.3.3 Transformation versus mapping

As already reported by for example Sim [2009], we hypothesize that phone mappings between phone sets adopted by different databases do not exist and expect the stochastic phone space transformation to outperform manual phone mappings as well as automatically, data-driven determined mappings.

We assume that the optimal phone mapping is a knowledge-driven manual mapping, i.e. mapping each target phone to the source phone that shares the same IPA symbol. For each target phone without a matching source phone, we manually select the most similar one according to the IPA chart, also given in Figure A.1, page 104. For information, the manual knowledge based mapping is given in Table A.2, page 103.

The estimation of $P(d_t^l|\boldsymbol{x},\Theta)$, as given in (3.6), page 33, is a weighted sum of all posterior estimates $P(s_t^k|\boldsymbol{x}_t,\Theta_S)$ (soft decision). Alternatively, a phone mapping takes a hard decision. i.e. just considers the most similar source phone. Similarly to PPM [Sim and Li, 2008], we also apply a *data-driven phone mapping*:

$$P(d_t^l|\boldsymbol{x}_t,\Theta) = P(s_t^{k^*}|\boldsymbol{x}_t,\Theta_S), \tag{3.24}$$

where the sum in (3.5), page 32, has been replaced by a max operator and where $k^* = \arg\max_k P(d^l|s^k,\Theta_M)$. Consequently, if the number of source phones (S) and the number of target phones (D) are different, we can distinguish between two cases:

- $D < S$: some source posteriors are discarded,
- $D > S$: multiple target phones are mapped to the same source class.

Both scenarios are suboptimal for decoding.

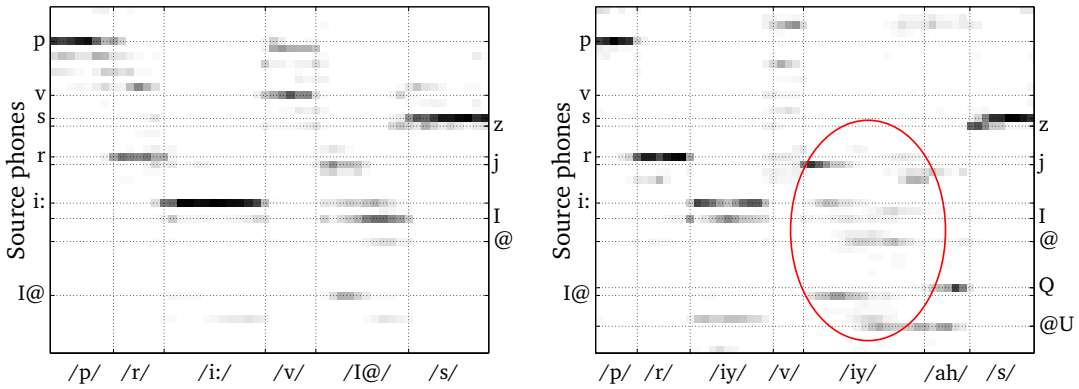
System	WACC	Accuracy of the utterance *previous*
Monolingual posterior transformation	93.3 %	84%
Data driven phone mapping	82.1 %	18%
Manual phone mapping	83.2 %	26%

Table 3.3: Comparison of monolingual phone space transformation and data driven as well as knowledge based phone mapping from SpechDat(II) English to HIWIRE non-native English (see Table A.2, page 103 for more the mapping details). WACC stands for word accuracy. The last column shows the accuracy of the one-word-utterance *previous*.

Monolingual transformation versus monolingual mapping

Table 3.3 shows that the monolingual stochastic transformation performs substantially better than the data-driven and knowledge based phone mapping on the HIWIRE test set. Earlier studies, performed on different datasets, also compared hard mapping (PPM) to soft mapping (PAM) and reported similar degradation (20% absolute increase in phone error rate) [Sim, 2009].

The HIWIRE database contains one-word commands as well as whole sentence utterances. An error analysis revealed that there is a considerable difference in the performance of decoding the one-word utterance *previous*, also given in Table 3.3. It appears 38 times in the HIWIRE test set. The stochastic mapping wrongly decodes it six times (84% accuracy) whereas the data-driven and knowledge based mapping yield 18% and 26% accuracy, respectively.



(a) Native speaker (b) Non-native speaker

Figure 3.3: Native and non-native posteriogram of the word *previous*. The transcriptions are given in SpeechDat(II) SAMPA format (native) and HIWIRE Arpabet format (non-native). The y-axis represents the British English source phone posteriors and is labeled on both sides for better readability. The chosen colormap represents 1.0 in black and 0.0 in white.

Figure 3.3 shows a typical British English source phone posteriogram of the word *previous* pronounced by a native (from the SpeechDat(II) data) and by a non-native (from the HIWIRE data). Note that the dictionaries of SpeechDat(II) and HIWIRE transcribe the word *previous*

differently. For the native speaker and the non-native speaker, the SpeechDat(II) and the HIWIRE variant are given, respectively, in Figure 3.3. Hence, the circled region in the non-native posteriogram is therefore modeled with the phoneme */iy/*. It is obvious from the figure, that a transformation outperforms a mapping in that region. Indeed the probabilities $P(d^\ell = /iy/|s^k)$ of the stochastic transformation are displayed in Figure 3.4 and it can be seen that many source phone posteriors contribute to the target posterior of */iy/*. If phone mapping is applied on the other hand, only the source phone with the maximum probability, */i:/* in this case, is considered.

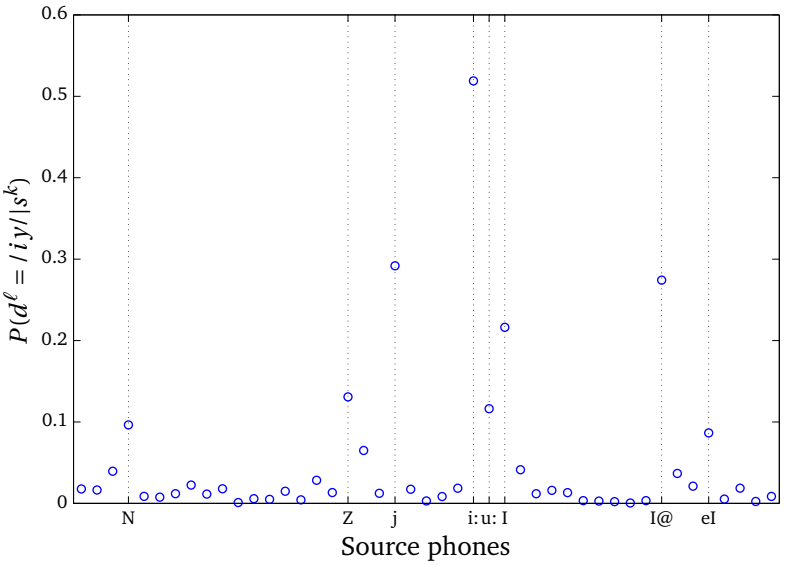


Figure 3.4: Stochastic parameters for the Arpabet phoneme */iy/*. Multiple SAMPA phonemes of the English SpeechDat(II) phoneme set are considered during the target phone posterior estimation.

Multilingual transformation versus multilingual mapping

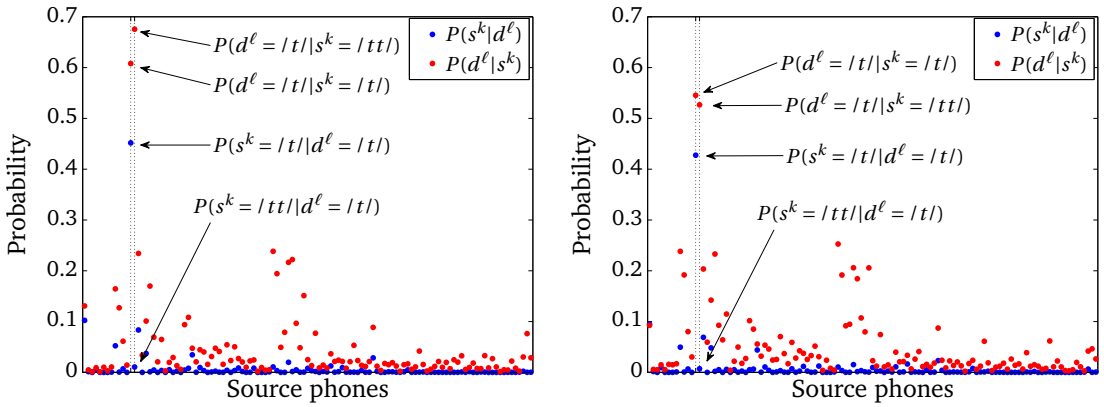
Similar to the monolingual transformation and mapping experiments, we can also perform multilingual transformation and mapping experiments. The data driven mapping and the posterior transformation are obtained based on the multilingually trained MLPs, MLP-UNI and MLP-sUNI. For the manual phone mapping, we map each target phone from the HIWIRE phone set to the universal phone that shares the same IPA symbol (adaptation data is not used). For each target phone without a matching source phone, we manually select the most similar one according to the IPA chart. The manual mapping is given in Table A.2, page 103.

Table 3.4 shows the results. Interestingly, we note that the performance of MLP-UNI is worse than the performance of MLP-sUNI if we apply data-driven phone mapping. This may result from the fact that larger MLPs (like MLP-UNI) will be more *discriminant*, yielding much lower probabilities to rare phone classes such as */nn/, /pp/, /bb/, /tt/, /dd/* (see Table A.2, page 103). In those cases, the denominator of (3.6), page 33, $P(s^k|\Theta_M)$, tends to dominate the numerator.

Table 3.4: Comparison of multilingual phone space transformation and data driven as well as knowledge based phone mapping on English non-native data (see Table A.2, page 103 for more the mapping details). MLP-UNI is trained on 63 h of data and MLP-sUNI on 12.7 h.

System	MLP	Word accuracy
Multilingual posterior transformation	MLP-UNI	96.0 %
Multilingual posterior transformation	MLP-sUNI	94.3 %
Data driven phone mapping	MLP-UNI	61.7 %
Data driven phone mapping	MLP-sUNI	69.4 %
Manual phone mapping	MLP-UNI	87.2 %
Manual phone mapping	MLP-sUNI	81.2 %

As a result, those rare phones will be more often used for the data-driven phone mapping. A comparison of the data-driven phone mappings of MLP-UNI and MLP-sUNI, shown in Table A.2, page 103, confirms that the mappings mostly differ for consonants like */n/*, */p/*, */b/*, */t/*, */d/*. Additionally, Figure 3.5 displays the estimates of $P(s^k|d^\ell)$ and $P(d^\ell|s^k)$ for the destination phone */t/*. It can be seen that $P(d^\ell = /t/|s^k = /ttt/)$ is higher than $P(d^\ell = /t/|s^k = /t/)$ for system MLP-UNI, but not for system MLP-sUNI.



(a) MLP-UNI (b) MLP-sUNI

Figure 3.5: Comparison of the estimates of $P(s^k|d^\ell)$ and $P(d^\ell|s^k)$ for the systems MLP-UNI and MLP-sUNI. The conditional probability $P(d^\ell = /t/|s^k = /ttt/)$ is higher than $P(d^\ell = /t/|s^k = /t/)$ in the case of system MLP-UNI because of the low prior $P(s^k = /ttt/)$.

3.3.4 Transformation versus full system training

Instead of training the stochastic transformation matrix Θ_M on the HIWIRE adaptation data, we can also train a full hybrid system on the adaptation data. Firstly, this implies to train an MLP on the accented English adaptation set (MLP-AE) to estimate target phone class posteriors $P(d_t^\ell|\boldsymbol{x}_t)$ directly. During MLP training, 90% of the adaptation data is used for training and the remaining 10% for validation. The training of an MLP requires frame-based alignments. However, no alignments are available for HIWIRE. Therefore, we perform forced

Table 3.5: Summary of the hybrid system trained on HIWIRE data. The total amount of training data, the frame accuracy on the development data, the source set including the number of phones (S) is given as well as the WACC on the test set is given.

System	Source set	S	Training data	DEV frame accuracy	WACC
MLP-AE	Arpabet English	38	2.4 h	58.2%	92.8%

alignment with the best performing transformed models (MLP-UNI). Forced alignment is also used to estimate prior probabilities as needed in the hybrid system.

A summary of the hybrid system that is directly trained on the HIWIRE data is given in Table 3.5. It seems that the 149 min of adaptation data, provided by HIWIRE, is enough data to train a complete hybrid system. Therefore, we investigate smaller amounts of data to train the parameters of the stochastic transformation matrix Θ_M in the next section.

3.3.5 Dealing with small amount of training data

The number of parameters that need to be estimated for the stochastic transformation is relatively small. In our case, the size of the stochastic mapping matrix is $S \times D$, S being the number of source phones and D the number of target phones, i.e. 117 x 38 for MLP-UNI. Hence, we expect the proposed approach to perform well even for very small amounts of data. To confirm that hypothesis, we continuously decrease the amount of available data, by considering fewer utterances per speaker as seen in Table 3.6. For these experiments, we always use system MLP-UNI because it performed best in previous experiments.

To have at least one acoustic sample for each target phone, we can not consider all speakers anymore for datasets of less than ten minutes duration. The 3-minutes dataset is obtained by the following heuristic: beginning with the list of files from the 32-minutes dataset and including an utterance if it contains any phone not yet covered, otherwise discarding it. This procedure selects more utterances than necessary because frequent phones appear in many utterances. Therefore, for the 2-minutes dataset, we first sort the phones according to their frequency with the most rare phone first. For each unseen phone in the sorted phone list, we then include the first utterance of the 32-minutes dataset that contains it.

Table 3.6: Utterance choice on the HIWIRE data to simulate low amount of data. Word accuracy performance of the multilingual phone space transformation (MLP-UNI) is also given.

Amount of data [min]	Considered Utterances	Word accuracy
149	Utterances 1-50	96.0%
32	Utterances 1-10	96.2%
10	Utterances 3,5,7	96.0%
3	Manually selected	95.1%
2	Manually selected	93.8%

Table 3.6 demonstrates the efficiency of the proposed approach through outstanding performance in the case of limited amounts of training data. However, it also shows that we are not able to take full advantage of the model in case of larger (typically more than 30 min) amounts of training data. Indeed, as already discussed at the beginning of this section, the investigated approach has a number of parameters equal to the size of the stochastic transformation matrix.

3.4 Conclusion

In this chapter, we have shown that different phone sets, associated with different databases, partition the same acoustic space differently and that manually derived phone mappings are detrimental to ASR systems. However, only ten minutes of adaptation data, along with phone transcriptions, are sufficient to transform multilingual source phone posterior probabilities to monolingual English phone posterior probabilities. The multilingual phone space transformation yields improvement on non-native ASR compared to the monolingual phone space transformation.

The parsimonious use of parameters makes the proposed system extremely efficient in terms of data requirement. On the other hand, the approach is not able to take full advantage of more than 30 min of adaptation data. The number of parameters in the stochastic transformation matrix could be increased if more reference posteriors are allowed.

Indeed, the current explicit transformation not only limits the approach in terms of parameters, but it also employs a different local score, as well as a different cost function, during the training of the reference posteriors, see (3.14), page 35 and Viterbi decoding, see (3.23), page 37. In the next chapter, we therefore introduce KL-HMM, an approach that uses the same KL divergence based local score during training and decoding and allows more flexibility in terms of modeling than the template-like approach presented in this chapter.

4 KL-HMM

The phone space transformation presented in the last chapter is limited in the number of parameters and utilizes different cost functions during training and decoding. Therefore, we replace the DTW based template-like training of the phone space transformation by a Viterbi-like HMM training that minimizes the same Kullback–Leibler (KL) divergence based cost function, and where the states of the HMM are parametrized with reference posteriors. Such an HMM, referred to as KL-HMM, can also be used for decoding and allows more flexibility in terms of modeling than the template-like approach presented in the last chapter.

Setting out from the point of view that ASR ought to benefit from data in languages other than the target language we revisit the recently proposed KL-HMM approach that is able to exploit multilingual information in the form of universal phone posterior probabilities conditioned on the acoustics. KL-HMM was first introduced by Aradilla [2008] and exploits an HMM where the states (hidden variables) are associated with the target phone sequence and an MLP that can be trained on source languages for which larger amounts of training data are available. To highlight these dependencies, the HMM is referred to as *target HMM* and the MLP is referred to as *source MLP*, respectively.

We extend the existing KL-HMM framework and formulate a means to train a context-dependent recognizer. Taking the Greek SpeechDat(II) data as an example, we show that the proposed formulation is sound. Furthermore, we show that it is able to outperform a current state-of-the-art HMM/GMM system in small amount of training data conditions. We also use a standalone Tandem system, as an additional reference point, and to further understand the properties of our system.

4.1 Model

In the simplest case, the target HMM uses one state per target phone d^{ℓ} in a *left-to-right* structure, which is obtained from the destination phone transcriptions. In Figure 4.1, for example, we consider the one-word utterance *previous*, as already done in Figure 3.2, page 33,

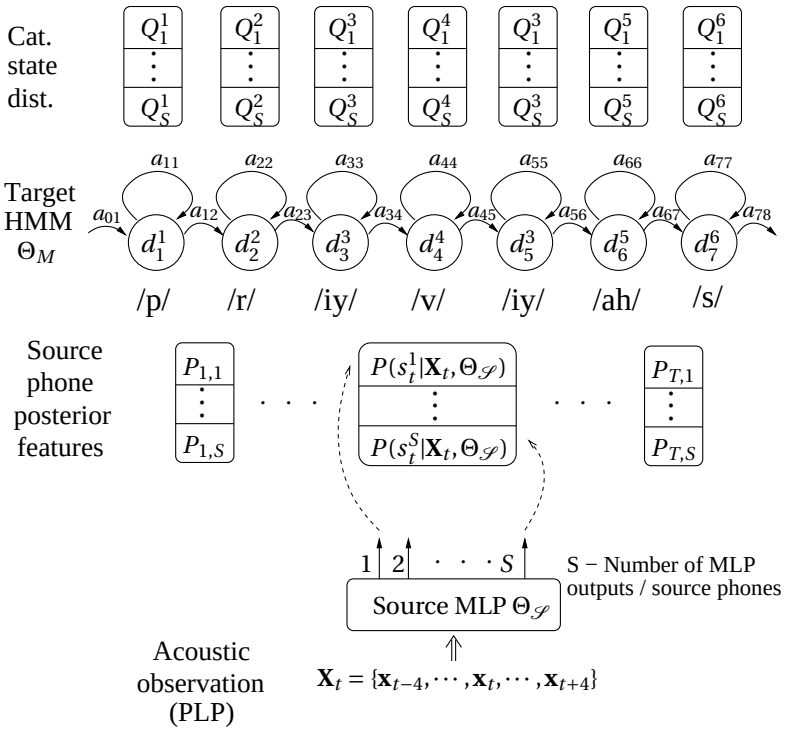


Figure 4.1: Illustration of the Kullback–Leibler divergence based HMM with categorical distributions $\boldsymbol{Q}^\ell$, target HMM Θ_M derived from the phonetical transcription and posterior features P_t obtained from the source MLP $\Theta_{\mathscr{S}}$.

resulting in the sequence of HMM states $\mathscr{D} = \{d_1^1, d_2^2, d_3^3, d_4^4, d_5^3, d_6^5, d_7^6\}$[1]. In this illustrative case, the associated HMM has seven states plus non-emitting start and end states. However, the presented algorithm is not limited to such simple HMM structures, but allows more complex ones such as using three states per phone. For the ease of notation, and without loss of generality, we limit ourselves to the simplest case (one state per phone) in the following derivations.

As seen in Figure 4.1, KL-HMM makes use of posterior features. In Section 2.3.1, we presented two different approaches based on posterior features, namely Tandem and hybrid systems. In a Tandem system, the emission probabilities associated with the states are modeled with a mixture of Gaussians. However, since the posterior features are not normally distributed, further processing in the form of logarithm and PCA is necessary. Hybrid systems on the other hand model the likelihood $p(\boldsymbol{x}_t|d^\ell)$ by converting posterior features using Bayes' rule. Even though posterior features are directly used, a hybrid system associates each HMM state with a particular output ℓ of the MLP and only makes use of the posterior probability $p(d^\ell|\boldsymbol{x}_t)$ instead of the whole probability vector.

[1] Note that an HMM sequence is written as $\mathscr{Q}$ in Chapter 2 because a state is usually denoted by q_t. In this section, we use $\mathscr{D}$ to highlight its relation with the destination phones and to avoid confusion with the categorical distributions $\boldsymbol{Q}^\ell$. Furthermore, the index n in d_n refers to the count of HMM states rather than time.

Indeed, KL-HMM is different from Tandem and hybrid systems, because it directly utilizes raw posterior features without further processing and each HMM state considers the whole probability vector rather than being bound to a single MLP output. More specifically, each target HMM state d_n^ℓ, with $\ell \in \{1, \cdots, D\}$ (D being the number of target phones) and $n = 1, \cdots, N$ (N being the total number of states of an HMM associated with an utterance), is thus parametrized by a categorical distribution $\boldsymbol{Q}^\ell$:

$$\boldsymbol{Q}^\ell = \boldsymbol{P}(\boldsymbol{s}|d^\ell, \Theta_M) = \begin{bmatrix} P(s^1|d^\ell, \Theta_M) \\ \vdots \\ P(s^S|d^\ell, \Theta_M) \end{bmatrix} = \begin{bmatrix} Q_1^\ell \\ \vdots \\ Q_S^\ell \end{bmatrix}.$$

A *categorical distribution* is a multinomial distribution where only one sample is drawn and can also be seen as a generalization of the Bernoulli distribution to more than two outcomes [Bishop, 2006, p.75]. Note that Bishop [2006] does not explicitly name such a distribution a categorical distribution. The categorical distributions of the HMM are essentially the same than the reference posteriors discussed in the Chapter 3. Therefore, we denote the parameter of the HMM by Θ_M and the dimensionality of $\boldsymbol{Q}^\ell$ is S, the total number of source phones. States d_3^3 and d_5^3 are parametrized with the same categorical $\boldsymbol{Q}^3$ because they are associated with the same target phone.

Of course, transition probabilities a_{ij}, to go from state i to state j, should also be parameters of the target HMM, $\Theta_M = \{\boldsymbol{Q}^\ell, a_{ij}\}$. However, we fix them to constant values of 0.5 (except for $a_{01} = 1$), as usually done in hybrid HMM/MLP systems.

4.2 Training

DTW as used in the previous chapter and the Viterbi algorithm used for HMM training are two instances of dynamic programming in which a global score is found as a sum of local distances along the optimal alignment between the input and reference [Aradilla, 2008]. Therefore, the KL divergence based local distance measure, used in the previous chapter, can be used as local score in a Viterbi-like segmentation optimization algorithm resulting in the following cost function:

$$\mathscr{F}(\mathscr{P}, \mathscr{Q}) = \sum_{t=1}^{T} d(\boldsymbol{P}_t, \boldsymbol{Q}^{\ell(\phi(t))}) \tag{4.1}$$

where $\ell(\phi(t))$ stands for the fact that ℓ is determined by ϕ, the path obtained from the segmentation step. During the segmentation step, also shown in Figure 4.2, the HMM aligns the observed posteriors (posterior features) with the states by minimizing the cost function, $\mathscr{F}(\mathscr{P}, \mathscr{Q})$, between the posterior feature sequence $\mathscr{P}$, and the categorical distributions $\mathscr{Q}$, associated with the HMM state sequence $\mathscr{D}$. Note that we omit the transition probabilities in (4.1) because we assume that they are constant. The optimization step is identical to the one presented in the last chapter, the arithmetic mean of all the posterior features associated

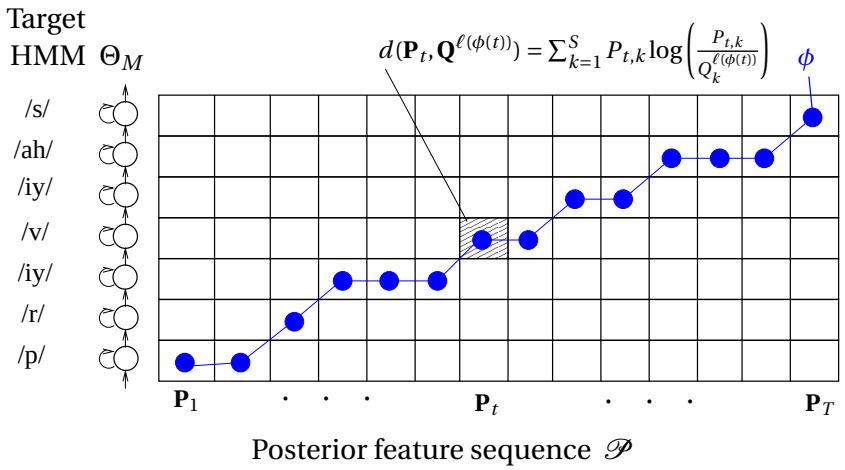


Figure 4.2: Segmentation step in the iterative Viterbi-like segmentation optimization algorithm. The segmentation ϕ can be optimized by aligning the posterior features with the target HMM Θ_M. During alignment, the cost function $\mathscr{F}(\mathscr{P},\mathscr{Q})$ is minimized.

with an HMM state parameterized with $\boldsymbol{Q}^{\ell}$:

$$Q_k^{\ell} = \frac{1}{T^{\ell}} \sum_{n:d_n^{\ell} \text{ in } \mathscr{D}} \sum_{t:\phi(t)=n} P_{t,k}, \tag{4.2}$$

where $\mathscr{D}$ stands for the sequence of HMM states associated with the training utterance and T^{ℓ} refers to the number of feature vectors associated with $\boldsymbol{Q}^{\ell}$.

4.3 Recognition

During HMM decoding (see also Chapter 2, page 15), we search for the optimal word sequence $\mathscr{W}^*$ that maximizes $P(\mathscr{W}|\boldsymbol{X}) \propto p(\boldsymbol{X}|\mathscr{W})P(\mathscr{W})$. The probability of a word sequence $P(\mathscr{W})$ is estimated by the language model, and $p(\boldsymbol{X}|\mathscr{W})$ can be written as:

$$p(\boldsymbol{X}|\mathscr{W}) = \sum_{\{\phi_{\mathscr{D}}\}^{\mathscr{W}}} p(\boldsymbol{X}|\phi_{\mathscr{D}})P(\phi_{\mathscr{D}}), \tag{4.3}$$

where $\{\phi_{\mathscr{D}}\}^{\mathscr{W}}$ denotes the set of all possible paths allowed by the HMM state sequence $\mathscr{D}$ that is dictated by the word sequence $\mathscr{W}$.

Using the Viterbi approximation, the sum in (4.3) can be replaced by the max operator:

$$p(\boldsymbol{X}|\mathscr{W}) \approx \max_{\{\phi_{\mathscr{D}}\}^{\mathscr{W}}} p(\boldsymbol{X}|\phi_{\mathscr{D}})P(\phi_{\mathscr{D}}). \tag{4.4}$$

and in the log domain:

$$p(\boldsymbol{X}|\mathscr{W}) \approx \max_{\{\phi_{\mathscr{D}}\}^{\mathscr{W}}} \left(\log p(\boldsymbol{X}|\phi_{\mathscr{D}}) + \log P(\phi_{\mathscr{D}})\right). \tag{4.5}$$

Table 4.1: Comparison of the multilingual phone space transformation to KL-HMM with different amounts of adaptation data (in minutes). Both systems use MLP-UNI to estimate phone posteriors. All the numbers stand for word accuracies on the HIWIRE test data.

Data (in minutes)	149	32	10	3	2
Multilingual posterior transformation	96.0%	96.2%	96.0%	95.1%	93.8%
KL-HMM	96.9%	96.7%	95.8%	94.7%	92.0%

As defined by Aradilla [2008, p. 94], the log-likelihood, $p(\boldsymbol{X}|\phi_{\mathscr{D}})$, can in the KL-HMM framework directly be replaced with the negative cost function, $-\mathscr{F}(\mathscr{P},\mathscr{Q})$:

$$p(\boldsymbol{X}|\mathscr{W}) \approx \max_{\{\phi_{\mathscr{D}}\}^{\mathscr{W}}} \left(-\mathscr{F}(\mathscr{P},\mathscr{Q}) + \log P(\phi_{\mathscr{D}})\right) = \min_{\{\phi_{\mathscr{D}}\}^{\mathscr{W}}} \left(\sum_{t=1}^{T} d(\boldsymbol{P}_t, \boldsymbol{Q}^{\ell(\phi_{\mathscr{D}}(t))}) - \log a_{ij}\right). \tag{4.6}$$

where the association of categorical distributions, $\boldsymbol{Q}^{\ell}$, to posterior feature, $\boldsymbol{P}_t$, is specified by the segmentation $\phi_{\mathscr{D}}(t)$, $i = \phi_{\mathscr{D}}(t-1)$ and $j = \phi_{\mathscr{D}}(t)$, respectively. Probabilistic grammars and word insertion penalties can be used in a similar way as in HMM/GMM based ASR systems [Aradilla, 2008].

4.4 Monophone KL-HMM

Similar to standard HMM/GMM systems, the most basic KL-HMM system makes use of monophones. As usually done, we model each monophone with three states. First, we compare such a monophone KL-HMM system to the posterior based stochastic phone space transformation evaluated in the last chapter. Since the HIWIRE database is limited in the amount of data, we then switch to a database with more than 10 h of training data to further investigate KL-HMM acoustic modeling.

4.4.1 KL-HMM versus posterior transformation on non-native ASR

To compare KL-HMM to the posterior transformations proposed in the Chapter 3, we build a monophone KL-HMM system with three states per phone. We used the multilingual MLP presented in Section 3.3.2, page 39, to extract posteriors features of dimension 117 and recognize non-native English speech from the HIWIRE dataset. As done for the phone space transformation, we also tune the word insertion penalty on the adaptation data. Table 4.1 compares the posterior based stochastic phone space transformation with KL-HMM.

KL-HMM performs better than the phone space transformation when there is more than 10 min of data (significant difference for 32 min and 149 min). If there is less, the transformation performs better. This can partially be explained by the fact that KL-HMM has three times more parameters because we model each phone with three states. However, the consistent usage of the KL divergence as a cost function during HMM training and decoding, and the

Table 4.2: KL-HMM monophone system performance on SpeechDat(II) Greek. All the numbers stand for word accuracies on the test set.

Data (in minutes)	5	9	18	37	75	152	308	808
KL-HMM mono	79.7%	79.9%	80.1%	80.2%	80.2%	80.4%	80.4%	80.5%

flexible HMM structure favors KL-HMM over the stochastic transformation. An extensive evaluation of KL-HMM on the HIWIRE data including mono-, cross- and multilingual setups and comparisons with related approaches is given in Chapter 5.

4.4.2 Boosting monolingual Greek ASR with multilingual resources

Although KL-HMM performs worse than a stochastic transformation if less than 10 min of data is available, results on HIWIRE suggest that KL-HMM has potential for under-resourced language ASR. In that context, we evaluate KL-HMM using SpeechDat(II) data from five European languages as available multilingual data (as already done for HIWIRE) and the Greek SpeechDat(II) database as representative of an unseen language with little available data.

The Greek SpeechDat(II) database contains a relatively large amount of data that we split into training (1,500 speakers), development (150 speakers) and testing (350 speakers) sets as done for the other SpeechDat(II) databases as well. To simulate limited resources, we continuously reduce the amount of available data by randomly picking a subset of utterances from the training set. The amount of training data varies from 13.5 hours to 5 minutes. There is only one global test set and all the systems, trained on different amounts of data, are evaluated on the same set. The test sentences use 10k different words and the dictionary makes use of 31 phonemes in SAMPA format, shown in Table A.1, page 102.

Since we have no access to an appropriate language model, we simply build two different language models: one with all the sentences from the development set, and one with all the sentences from the test set. Those language models have perplexities of 43 and 44, respectively. The development language model is used during the parameter tuning (language model scaling factor and word insertion penalty) on the development set and the test language model is used during the evaluation. In that sense, results should be considered as optimistic, but these experiments are anyway for the purpose of illustration only.

For the monophone KL-HMM experiments, we use the development set, to tune language model scaling factor and word insertion penalty for the system that uses 13.5 h of training data. We then fix these parameters to the same values during the subsequent decoding using smaller amounts of training data.

Results are given in Table 4.2. It can be observed that the systems yield similar results for 5 min and 808 min of training data. This can be attributed to the low number of parameters, $3 \times 31 \times 117$, three states per Greek phone multiplied with the dimension of the categorical

Table 4.3: KL-HMM triphone system performance on SpeechDat(II) Greek. The word accuracies on the test set, the relative performance change compared with the monophone KL-HMM system as well as the number of states is given. For unseen triphones in the test set, we back off to the monophone model of the corresponding center phone.

Data (in minutes)	5	9	18	37	75	152	308	808
KL-HMM tri	74.1%	76.3%	78.2%	80.0%	81.2%	82.4%	83.0%	83.4%
rel. +/- mono	-7.0%	-4.5%	-2.4%	-0.2%	1.2%	2.5%	3.2%	3.6%
Number of states	4,053	5,454	7,038	8,880	10,368	11,841	13,101	14,421

distributions. Therefore, in the next section, we investigate triphone based KL-HMMs.

4.5 Triphone KL-HMM

Similarly to standard HMM/GMM systems, triphone KL-HMM systems are built by extending the context of a monophone on the left and on the right by one, resulting in a triphone. In this section, we first limit ourselves to word-internal context-dependent triphone models. For unseen triphones in the test set, we back off to the monophone model of the corresponding center phone.

As done for the monophone KL-HMM system, we tune language model scaling factor and word insertion penalty only once on the development set for the system trained on all the training data.

Table 4.3 shows the performance of a triphone KL-HMM system and compares it to the monophone KL-HMM performance. Word accuracies, relative performance change of the triphone KL-HMM compared to the monophone KL-HMM system, as well as number of states of the corresponding acoustic models, are reported. The KL-HMM triphone system uses relatively large amounts of states, i.e. three states per triphone seen in the training set. Indeed, the triphone system yields improvement when at least 75 min of data are available for training. For small amounts of data, however, the monophone KL-HMM system still yields the best performance.

In creating triphone context models, we immediately run into the problem of sparsity of the training data, since many triphone contexts will occur infrequently or not at all. In standard HMM/GMM ASR systems, decision tree clustering approach [Young et al., 1994] was introduced in which states of context-dependent models are tied, thereby sharing data, according to shared properties. The state tying is performed by greedy optimization of a given criterion, usually maximum likelihood. An additional property of this approach is that it also permits the modeling of contexts that were unseen in the training data.

However, no such decision tree clustering algorithms have been available to date for the KL-HMM framework, i.e., in the context of posterior distributions. Therefore, in the next section,

we present an algorithm that allows us to perform decision tree clustering for KL-HMM based ASR systems.

4.6 Tied states KL-HMM

In this section, we first briefly present the standard likelihood based decision tree clustering before we introduce the novel algorithm for KL-HMMs. Then, we evaluate the proposed algorithm on the Greek SpeechDat(II) data and compare it to monophone and triphone KL-HMMs.

4.6.1 Likelihood based decision tree criterion

Suppose that we have a set of HMM states $\mathbb{D} = \{d^1, \cdots, d^D\}$, which we wish to tie using the standard decision tree method [Young et al., 1994] such that at the parent node we have a set of questions $\{m\}$. Each question can then split $\mathbb{D}$ into two non-overlapping subsets $\mathbb{D}_y(m)$ and $\mathbb{D}_n(m)$, where subscripts y and n indicate the binary split that separates the set into *yes* and *no* responses to question m. The questions and the tree topology are chosen to maximize the log-likelihood of the training data given the tied states (terminal nodes).

Assuming that 1) the assignments of observations to states are not altered during the clustering procedure, 2) the contribution of the transition probabilities to the total likelihood can be ignored, and 3) the total likelihood of the data can be approximated by a simple average of the log-likelihoods weighted by the probability of state occupancy, the log-likelihood of the training data can be approximated as [Young et al., 1994] :

$$\mathscr{L}(\mathbb{D}) \approx -\frac{1}{2}(\log[(2\pi)^C|\Sigma(\mathbb{D})|] + C) \sum_{d^\ell \in \mathbb{D}} \sum_{t=1}^{T} \gamma_\ell(\boldsymbol{x}_t), \tag{4.7}$$

where, for training data pooled in set of states $d^\ell \in \mathbb{D}$, $\mathscr{L}(\mathbb{D})$ is the log-likelihood, $\Sigma(\mathbb{D})$ is the variance of data in the set of states $\mathbb{D}$, T is the number of frames in the training data and $\gamma_\ell(\boldsymbol{x}_t)$ is the posterior probability of state d^ℓ for acoustic observation vector $\boldsymbol{x}_t$ of dimension C. Assuming hard occupation decision for states, i.e. $\tilde{\ell} = \arg\max_\ell \ \gamma_\ell(\boldsymbol{x}_t) : \ \gamma_{\tilde{\ell}} = 1, \gamma_{\ell \neq \tilde{\ell}} = 0$, we can further simplify (4.7):

$$\mathscr{L}(\mathbb{D}) \approx -\frac{1}{2}(\log[(2\pi)^C|\Sigma(\mathbb{D})|] + C) \sum_{d^\ell \in \mathbb{D}} T^\ell, \tag{4.8}$$

where T^ℓ is the number of times that state d^ℓ is observed in the training data.

Since each question splits $\mathbb{D}$ into two non-overlapping subsets $\mathbb{D}_y(m)$ and $\mathbb{D}_n(m)$ at each node, we can choose the question m that maximizes the likelihood difference $\Delta\mathscr{L}(m|\mathbb{D})$:

$$\Delta\mathscr{L}(m|\mathbb{D}) = \mathscr{L}(\mathbb{D}_y(m)) + \mathscr{L}(\mathbb{D}_n(m)) - \mathscr{L}(\mathbb{D}). \tag{4.9}$$

To avoid overfitting, the stopping criterion is usually based on a combination of minimum cluster occupancy and minimum increase in log-likelihood threshold. The latter can automatically be determined with the minimum description length (MDL) criterion [Shinoda and Watanabe, 1997, Zen et al., 2007].

It is evident from these equations that the likelihood does not depend on the training observations themselves but merely on the variance over training data corresponding to the states (which can be calculated from the state probability density functions) and the state occupancy statistics. Next, we show that a similar derivation exists for systems that use a KL divergence based cost function to perform ASR.

4.6.2 Kullback–Leibler divergence based decision tree criterion

The goal of this section is to derive a decision tree clustering algorithm that is based on the KL divergence and independent of the posterior features $\boldsymbol{P}_t$. The proposed approach is similar in spirit to the decision tree clustering approach that uses the entropy based distance measure [e.g. Rogina, 1997]. The KL divergence is not symmetric and Aradilla [2008] derived training algorithms for both asymmetric KL divergence based local scores. The state distribution estimates resulting from $d(\boldsymbol{P}_t, \boldsymbol{Q}^\ell)$ as defined in (3.14), page 35, is given in (4.10). Since $d(.)$, as defined in (3.14), is not symmetric, the state distribution estimates for $d(\boldsymbol{Q}^\ell, \boldsymbol{P}_t)$ are different and given in (4.11).

$$Q_k^\ell = \frac{1}{T^\ell} \sum_{t=1}^{T^\ell} P_{t,k}, \tag{4.10}$$

$$Q_k^\ell = \frac{\tilde{Q}_k^\ell}{||\tilde{\boldsymbol{Q}}^\ell||_1}, \tag{4.11}$$

with

$$\tilde{Q}_k^\ell = \left(\prod_{t=1}^{T^\ell} P_{t,k} \right)^{\frac{1}{T^\ell}}, \tag{4.12}$$

and $||.||_1$ being the L^1 norm:

$$||\tilde{\boldsymbol{Q}}^\ell||_1 = \sum_{k=1}^{S} \tilde{Q}_k^\ell. \tag{4.13}$$

Note that the sum and the product in (4.10) and (4.12) are over the T^ℓ feature vectors that are associated with state d^ℓ. Hence, as we have already seen, $d(\boldsymbol{P}_t, \boldsymbol{Q}^\ell)$ leads to the arithmetic mean. The local score $d(\boldsymbol{Q}^\ell, \boldsymbol{P}_t)$ on the other hand, leads to the normalized geometric mean. The choice to use $d(\boldsymbol{P}_t, \boldsymbol{Q}^\ell)$ for HMM training and decoding instead of $d(\boldsymbol{Q}^\ell, \boldsymbol{P}_t)$ is discussed later in Chapter 5. One of the main properties of the standard likelihood based decision tree clustering algorithm is that the likelihood does not depend on the training observations

themselves. For the KL divergence based decision tree clustering algorithm we therefore use $d(\boldsymbol{Q}^\ell, \boldsymbol{P}_t)$ because there is no closed form solution that is independent of the observed posteriors $\boldsymbol{P}_t$ for the proposed algorithm if $d(\boldsymbol{P}_t, \boldsymbol{Q}^\ell)$ is used.

The local score $d(\boldsymbol{Q}^\ell, \boldsymbol{P}_t)$ is always positive, and zero if and only if the posterior feature vector and the state posterior vector are the same, i.e.:

$$d(\boldsymbol{Q}^\ell, \boldsymbol{P}_t) \geq 0 \text{ and } d(\boldsymbol{Q}^\ell, \boldsymbol{P}_t) = 0 \text{ iff } \boldsymbol{P}_t = \boldsymbol{Q}^\ell. \tag{4.14}$$

Hence, instead of maximizing the log-likelihood, we propose to minimize:

$$\mathcal{K}(\mathbb{D}) = \sum_{d^\ell \in \mathbb{D}} \sum_{t=1}^{T^\ell} \sum_{k=1}^{S} Q_k^{\mathbb{D}} \log\left(\frac{Q_k^{\mathbb{D}}}{P_{t,k}}\right), \tag{4.15}$$

where $\mathbb{D}$ is a set of states d^ℓ and $\boldsymbol{Q}^{\mathbb{D}}$ the categorical distribution associated with a set of states. In the reminder of this section we show that $\boldsymbol{Q}^{\mathbb{D}}$ can be obtained based on the individual state distributions $\boldsymbol{Q}^\ell$ independently of $\boldsymbol{P}_t$ and subsequently formulate $\mathcal{K}(\mathbb{D})$ only dependent on $\boldsymbol{Q}^\ell$, $||\tilde{\boldsymbol{Q}}^\ell||_1$ and T^ℓ.

Categorical distribution associated with a set of states $\boldsymbol{Q}^{\mathbb{D}}$

Given (4.11), $Q_k^{\mathbb{D}}$, can be written as:

$$Q_k^{\mathbb{D}} = \frac{\tilde{Q}_k^{\mathbb{D}}}{||\tilde{\boldsymbol{Q}}^{\mathbb{D}}||_1}, \tag{4.16}$$

with

$$\tilde{Q}_k^{\mathbb{D}} = \left(\prod_{d^\ell \in \mathbb{D}} \prod_{t=1}^{T^\ell} P_t\right)^{\frac{1}{\sum_{d^\ell \in \mathbb{D}} T^\ell}} = \left(\prod_{d^\ell \in \mathbb{D}} \left(\tilde{Q}_k^\ell\right)^{T^\ell}\right)^{\frac{1}{\sum_{d^\ell \in \mathbb{D}} T^\ell}} = \left(\prod_{d^\ell \in \mathbb{D}} \left(Q_k^\ell ||\tilde{\boldsymbol{Q}}^\ell||_1\right)^{T^\ell}\right)^{\frac{1}{\sum_{d^\ell \in \mathbb{D}} T^\ell}}. \tag{4.17}$$

Hence, we can express $\boldsymbol{Q}^{\mathbb{D}}$ based on $\boldsymbol{Q}^\ell$, $||\tilde{\boldsymbol{Q}}^\ell||_1$ and T^ℓ, thus without having access to the posterior features $\boldsymbol{P}_t$.

KL divergence based decision tree cost function $\mathcal{K}(\mathbb{D})$

The KL divergence based decision tree cost function given in (4.15) can be expanded as:

$$\mathcal{K}(\mathbb{D}) = \sum_{d^\ell \in \mathbb{D}} \sum_{t=1}^{T^\ell} \sum_{k=1}^{S} Q_k^{\mathbb{D}} \log Q_k^{\mathbb{D}} - \sum_{d^\ell \in \mathbb{D}} \sum_{t=1}^{T^\ell} \sum_{k=1}^{S} Q_k^{\mathbb{D}} \log P_{t,k}, \tag{4.18}$$

$$= \sum_{d^\ell \in \mathbb{D}} T^\ell \sum_{k=1}^{S} Q_k^{\mathbb{D}} \log Q_k^{\mathbb{D}} - \sum_{k=1}^{S} Q_k^{\mathbb{D}} \sum_{d^\ell \in \mathbb{D}} \sum_{t=1}^{T^\ell} \log P_{t,k}, \tag{4.19}$$

where (4.19) exploits the fact that $Q_k^{\mathbb{D}}$ does not depend on t. Furthermore, the second term of (4.19) can be simplified as follows:

$$\sum_{k=1}^{S} Q_k^{\mathbb{D}} \sum_{d^\ell \in \mathbb{D}} \sum_{t=1}^{T^\ell} \log P_{t,k} = \sum_{k=1}^{S} Q_k^{\mathbb{D}} \log\left(\prod_{d^\ell \in \mathbb{D}} \prod_{t=1}^{T^\ell} P_{t,k} \right), \tag{4.20}$$

$$= \sum_{k=1}^{S} Q_k^{\mathbb{D}} \sum_{d^\ell \in \mathbb{D}} T^\ell \log\left(\tilde{Q}_k^{\mathbb{D}}\right), \tag{4.21}$$

$$= \sum_{k=1}^{S} Q_k^{\mathbb{D}} \sum_{d^\ell \in \mathbb{D}} T^\ell \log\left(Q_k^{\mathbb{D}} || \tilde{\boldsymbol{Q}}^{\mathbb{D}} ||_1\right), \tag{4.22}$$

$$= \sum_{k=1}^{S} Q_k^{\mathbb{D}} \sum_{d^\ell \in \mathbb{D}} T^\ell \left(\log\left(Q_k^{\mathbb{D}}\right) + \log\left(|| \tilde{\boldsymbol{Q}}^{\mathbb{D}} ||_1\right)\right). \tag{4.23}$$

Substituting (4.23) into (4.19) yields:

$$\mathcal{K}(\mathbb{D}) = \sum_{d^\ell \in \mathbb{D}} T^\ell \sum_{k=1}^{S} Q_k^{\mathbb{D}} \log Q_k^{\mathbb{D}} - \sum_{d^\ell \in \mathbb{D}} T^\ell \sum_{k=1}^{S} Q_k^{\mathbb{D}} \left(\log\left(Q_k^{\mathbb{D}}\right) + \log\left(|| \tilde{\boldsymbol{Q}}^{\mathbb{D}} ||_1\right)\right), \tag{4.24}$$

$$= - \sum_{d^\ell \in \mathbb{D}} T^\ell \log\left(|| \tilde{\boldsymbol{Q}}^{\mathbb{D}} ||_1\right) \sum_{k=1}^{S} Q_k^{\mathbb{D}}, \tag{4.25}$$

$$= - \sum_{d^\ell \in \mathbb{D}} T^\ell \log\left(|| \tilde{\boldsymbol{Q}}^{\mathbb{D}} ||_1\right), \tag{4.26}$$

since by definition: $\sum_{k=1}^{S} Q_k^{\mathbb{D}} = 1$.

Combining (4.13), (4.17) and (4.26) leads to:

$$\mathcal{K}(\mathbb{D}) = - \left(\sum_{d^\ell \in \mathbb{D}} T^\ell \right) \log\left(\sum_{k=1}^{S} \left(\prod_{d^\ell \in \mathbb{D}} \left(Q_k^\ell || \tilde{\boldsymbol{Q}}^\ell ||_1 \right)^{T^\ell} \right)^{\frac{1}{\sum_{d^\ell \in \mathbb{D}} T^\ell}} \right). \tag{4.27}$$

Thus, the KL divergence based decision tree cost function, $\mathcal{K}(\mathbb{D})$, can be calculated based on the statistics $\boldsymbol{Q}^\ell$, $||\tilde{\boldsymbol{Q}}^\ell||_1$, and T^ℓ of the individual states.

For the splitting of a set of states $\mathbb{D}$, we propose to choose the question m that maximizes the KL divergence based cost function difference $\Delta\mathcal{K}(m|\mathbb{D})$:

$$\Delta\mathcal{K}(m|\mathbb{D}) = \mathcal{K}(\mathbb{D}) - \left(\mathcal{K}(\mathbb{D}_y(m)) + \mathcal{K}(\mathbb{D}_n(m))\right), \tag{4.28}$$

in order to minimize $\mathcal{K}$. Similarly to the likelihood based decision tree, the stopping criteria can be based on a combination of minimum cluster occupancy and minimum decrease in the cost function threshold. For the likelihood based tree, the MDL criterion can be used to determine the minimum increase in log-likelihood threshold automatically. However, it is not evident how to determine the minimum description length for a posterior based model such as KL-HMM. Therefore, in this thesis, we tune the minimum decrease in the cost function

threshold on a development set rather than determining it automatically.

In a multilingual setup, there can be a mismatch between presented context in the multilingual decision tree and the context in the new target language. To address this issue, a polyphone decision tree specialization algorithm was proposed [Schultz and Waibel, 2000]. In our case however, the multilingual data is used to train the MLP and we use only Greek data to build the decision tree. Hence, this alleviates the need of such an algorithm.

4.6.3 Comparison of monophone, triphone and tied states KL-HMM

We evaluate the proposed decision tree clustering algorithm on Greek SpeechDat(II) data. As usually done in HMM/GMM ASR systems, for KL-HMM, we also build one decision tree for each phone [Young et al., 2006]. The root node of the tree contains all the triphone models where the corresponding phone is the center phone. Then, the tree is built according to the procedure described in Section 4.6.2. We fix the the minimum occupancy threshold to 20. Similar to the other systems presented in this chapter, the language model scaling factor, the word insertion penalty and the minimum decrease in the cost function threshold (decision tree) are tuned only once on the development set for the system trained on all the training data. For the successive experiments on subsets of the training data, the same values are used. Word accuracies and the number of tied states of the corresponding acoustic model are given in Table 4.4.

Figure 4.3 shows the performance of the monophone KL-HMM (KL-HMM mono), the triphone KL-HMM with backoff strategy (KL-HMM tri) as well as the KL-HMM with tied states (KL-HMM tied). The tied states KL-HMM system is always the best performing system. For only 5 min of data it performs marginally better than KL-HMM mono, but for 9 min of data it already performs significantly better. The tied states KL-HMM system performs significantly better than KL-HMM tri for all investigated amounts of training data.

4.7 Comparison of KL-HMM, MLLR, MAP and Tandem

As additional reference points, the results of a standard HMM/GMM system, a maximum likelihood linear regression (MLLR) system, a maximum a posteriori (MAP) adaptation system, and a multilingual standalone Tandem system are compared to tied states KL-HMM in Figure 4.4. Note that the scale of the y-axis is not the same for Figures 4.3 and 4.4. For each

Table 4.4: KL-HMM tied states system performance on SpeechDat(II) Greek. The word accuracies on the test set as well as the number of tied states is given.

Data (in minutes)	5	9	18	37	75	152	308	808
KL-HMM tied	80.3%	81.1%	82.2%	83.0%	83.6%	84.0%	84.4%	84.8%
Number of states	110	133	180	270	440	734	1,213	2,278

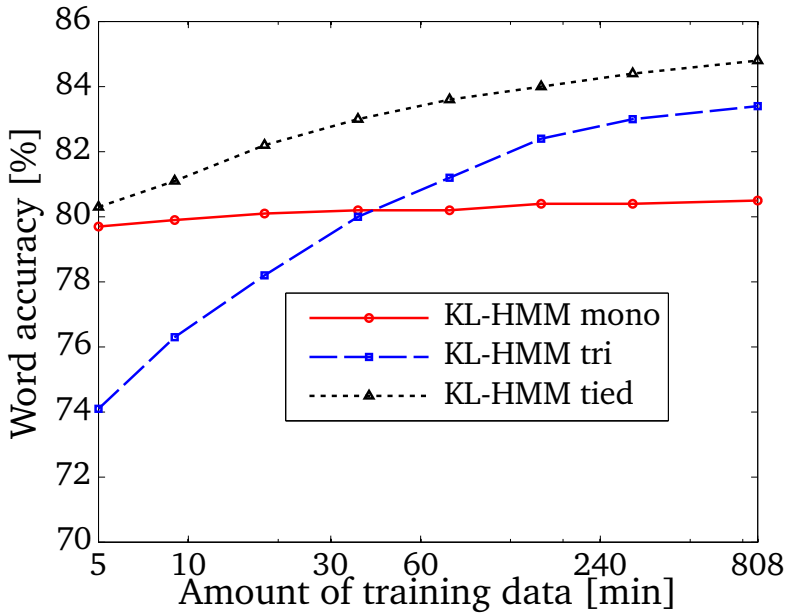


Figure 4.3: Comparison of the monophone, triphone and tied states KL-HMM systems on Greek ASR, using different amounts of training data. The x-axis is in logarithmic scale.

acoustic modeling technique, language model scaling factor and word insertion penalty are tuned for the system that uses all the training data.

The HMM/GMM system uses MF-PLP features and the MDL based decision tree clustering [Shinoda and Watanabe, 1997]. The tied triphone models are then modeled with 2, 4, 8 and 16 mixtures of Gaussians with diagonal covariance. Rather than using sophisticated algorithms such as *split and merge* [Ueda et al., 1998], we limit ourselves to using the same number of Gaussians for all states, but tune that number on development data. The MDL based decision tree clustering leads to 2,817 states each modeled with a mixture of 16 Gaussians for the system that uses all the training data and 107 states each modeled with a mixture of 4 Gaussians for the system that uses 5 min of data. Hence, the standard likelihood based decision tree clustering algorithm and the KL divergence based novel decision tree clustering algorithm yield similar number of states. However, the total number of parameters differ substantially because of the parsimonious use of parameters of the KL-HMM system (117 per state) compared to a GMM system with for example 4 Gaussians ($4 \times 39 + 4 \times 39 + 3 = 315$ per state).

If all the training data is used, there is only a marginal, insignificant, difference between the performance of the standard HMM/GMM ASR system and the tied states KL-HMM system. For the systems that are trained on less than 1 h of training data, the tied states KL-HMM system performs significantly better than the HMM/GMM system. However, the HMM/GMM system does not use the multilingual data that the KL-HMM indirectly uses through the MLP trained on the five European languages British English, Italian, Spanish, Swiss French and Swiss German.

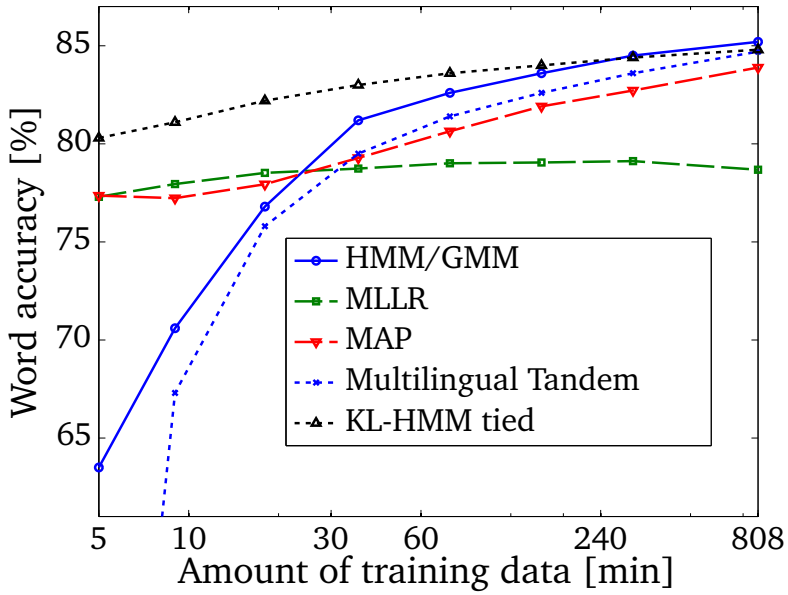


Figure 4.4: Comparison of tied states KL-HMM, HMM/GMM, MLLR, MAP adaptation and multilingual standalone Tandem on Greek ASR, using different amounts of training data. The x-axis is in logarithmic scale.

Therefore, we also evaluate the multilingual Tandem system that uses conventional HMM/GMM structures to model the universal posterior features. Besides the choice of the features, the training is same as for the monolingual standard HMM/GMM system. To model universal posteriors with Gaussians, as usually done, we apply logarithm and PCA to decorrelate. To directly compare the impact of the different modeling techniques (GMMs versus KL-HMM), we evaluate a standalone Tandem system and do not concatenate the MF-PLP features as often done. For systems trained form less than 2 h of training data, the tied states KL-HMM system yields significantly better results than the multilingual Tandem system. The results, shown in Figure 4.4, suggest that the multilingual Tandem system is suitable to only a limited extent for exploiting the multilingual information. However, for the experiments reported here, we keep all the dimensions after PCA, hence the number of parameters that need to be trained on small amounts of data are considerably higher than for the KL-HMM system. We will see later in Chapter 6 how to use Tandem systems more efficiently.

Furthermore, to evaluate whether the new language can be accommodated by linear transforms, we first train a triphone HMM/GMM system on the multilingual data (using the universal phone set). Each triphone state is modeled with a mixture of 16 Gaussians. Then, we apply the standard MLLR and use a regression tree that allows up to 32 regression classes to adapt the universal phone models to the target language. Since not all the Greek phones are present in the universal phone set, we map the palatal plosives c and ɟ to the velar plosives k and g respectively. However, as seen in Figure 4.4, the MLLR performance on this language adaptation task is rather low.

From earlier studies, we know that MAP adaptation performs better than MLLR if there is more than about 10 min of data [Wang et al., 2003]. Therefore, we also investigate MAP adaptation. We use the same multilingual seed models that we already used for the MLLR experiments and apply standard MAP adaptation. Hence, the mean $\boldsymbol{\mu}_m^\ell$ of mixture component m and state ℓ is adapted as follows:

$$\hat{\boldsymbol{\mu}}_m^\ell = \frac{N_m^\ell}{N_m^\ell + \tau}\boldsymbol{\mu}_m^{G,\ell} + \frac{\tau}{N_m^\ell + \tau}\boldsymbol{\mu}_m^{M,\ell}, \tag{4.29}$$

where N_m^ℓ is the occupation likelihood of the Greek data, τ a parameter to tune, $\boldsymbol{\mu}^G$ the mean of the Greek data and $\boldsymbol{\mu}^M$ the mean of the multilingual data. We tune τ for each system and apply the same manual phone mapping as done for MLLR. As expected, Figure 4.4 shows that MAP adaptation performs better than MLLR for larger datasets (more than 30 min of data). For smaller datasets, the performance of MLLR and MAP adaptation is not statistically different.

Altogether, the tied states KL-HMM system yields the best performance. Note that similar figures can be found in [Imseng et al., 2012b] and [Imseng et al., 2012d]. For those experiments, we used a symmetric KL divergence based measure. However, more recently we found that the asymmetric KL divergence is in fact more robust. This is also intuitively reasonable in that the underlying acoustic modeling problem is not symmetric since we observe the posterior features and train the categorical distributions. This effect is more pronounced for tasks, where the source and target phone sets essentially differ. We will further discuss the choice of the local score measure later in Section 5.4.

4.8 Conclusion

In this chapter, we extended the existing KL-HMM framework and presented a decision tree state clustering algorithm for KL-HMM systems. For the evaluation, we used multilingual data from five source languages to boost the performance of a Greek speech recognizer and simulated low-resource scenarios by restricting the amount of Greek training data.

The tree-based KL-HMM system successfully exploits multilingual information in the form of universal phone posterior features and outperforms all other systems for very small amounts of data (less than one hour). If there are 10 h of training data available, there is no statistically significant difference between the performance of the tied states KL-HMM system and a standard HMM/GMM system.

In the next two chapters, we will further apply the tied states KL-HMM system to non-native and under-resourced language ASR, and extensively compare it to related approaches.

5 Non-native ASR

In this chapter, we apply KL-HMM to non-native speech recognition. We start with the hypothesis that KL-MM is beneficial for non-native and accented speech because we can train the source MLP ($\Theta_{\mathscr{S}}$) with large amounts of multilingual data and then handle the variability in pronunciations with small amounts of data by learning the KL-HMM parameters Θ_M.

After a review of related work, we first explore multilingual KL-HMM systems and apply them to the non-native HIWIRE dataset. Therefore, we use MLP-UNI, the MLP trained on multilingual data as explained in Section 4.4.1, page 51, where we reported stochastic phone space transformation and monophone KL-HMM performance on the HIWIRE dataset. In this section, we extend the study to a tied states KL-HMM system.

Then, instead of using multilingual MLPs, we also investigate MLPs trained on a language different from the target language (crosslingual KL-HMM). We train four different MLPs on Italian, Spanish, Swiss French and Swiss German and subsequently use them as posterior feature estimators. We then compare the performance of the crosslingual systems on recordings of speakers with a non-native accent. We show that an MLP trained on out-of-language data is beneficial for the ASR performance on non-native data. This effect gets more pronounced if the MLP is trained on data from the mother tongue of a non-native speaker.

Furthermore, we conclude the studies on non-native speech with an extensive theoretical and experimental comparison of KL-HMM to related approaches.

5.1 Related work

We discussed earlier, in Chapter 3, that humans are able to produce a large variety of acoustic sounds which linguists have categorized into segments called phones, and that all those phones, across speakers and languages, share a common acoustic space. It is also known that acoustic realizations of the same phone exhibit high variability. Modeling variability of the acoustic realizations becomes even more challenging if we have to deal with non-native speech, because often phone realizations from two different languages are borrowed [Van Com-

pernolle, 2001]. Therefore, we hypothesize that KL-HMM, which is able to model the relation between different phones through implicit stochastic phone transformations, is beneficial for non-native ASR. Hereafter, we review related approaches that also model the relation between different phones.

Similarly to semi-continuous HMMs (SCHMMs), and also presented in Chapter 2, page 16, Schultz and Waibel [2001] proposed ML-tag, an HMM-based method to estimate language-independent acoustic models. In a conventional HMM/GMM framework, each state is modeled with a mixture of Gaussian distributions. If the IPA symbol set of two context-dependent states from different languages is the same, the training data of all involved languages is then used for the estimation of the Gaussian components (means and variances). The mixture weights, however, are trained for each language individually. The universal phone model is then transformed to a language specific model by estimating language dependent weights. In contrast to ML-tag, our work focuses on hybrid HMM/MLP systems and not on HMM/GMM systems, but we will show in Section 5.4.1 that KL-HMM is closely related to conventional Gaussian mixture based SCHMM systems.

Sim and Li [2008] proposed explicit one-to-one probabilistic phone mapping (PPM) that makes use of explicit phonetic reference transcriptions (in the form of target phones) and outputs of a phone recognizer that uses source phones. As a result, PPM maps each target phone to the most similar source phone. Then, Sim [2009] extended PPM to probabilistic acoustic mapping (PAM) for hybrid HMM/MLP ASR systems, which allows implicit posterior transformations. KL-HMM is similar in spirit to PAM. Both approaches are based on posterior space transformations, and we compare them in detail in Section 5.4.2.

Similarly to PAM, hidden feature transformation [Gemello et al., 2007] can be used to improve non-native ASR. More specifically, in a hybrid HMM/MLP framework, a linear transformation is applied to the activation of an internal layer of the MLP. The transformation is performed with a linear hidden network (LHN), which is trained with the standard MLP error back-propagation algorithm. However, since a hidden layer is adapted, LHN is bound to a fixed phoneme set and therefore less flexible than PAM and KL-HMM. We compare KL-HMM to LHN in Section 5.4.3.

Various studies applied acoustic model transformations to non-native ASR in the form of conventional adaptation techniques such as MLLR [Gales, 1998, Segura et al., 2007] or MAP [Gauvain and Lee, 1993, Wang et al., 2003].

More recently, combining acoustic model transformation and pronunciation modeling for non-native ASR was also investigated [Bouselmi et al., 2012]. For acoustic model transformation, MAP and model re-estimation were evaluated and combined with pronunciation modeling that was based on phonetic rule extraction. The phonetic rules were extracted by comparing the canonical transcription to the transcription given by a phonetic recognizer. However, if the mother tongue of the (non-native) speaker was unknown, MAP and model re-estimation alone performed better than in combination with pronunciation modeling.

5.2 Multilingual KL-HMM

The multilingual KL-HMM system uses the posterior features, estimated by MLP-UNI, the universal MLP, introduced in Section 3.3.2, page 39, and briefly recalled hereafter in the experimental setup section. Section 5.2.2 then presents the results. Note that we already presented results of the multilingual posterior transformation and the multilingual monophone KL-HMM in Table 4.1, page 51. In this section, we also investigate a tied states KL-HMM system, that exploits multilingual posteriors, on the non-native English database, HIWIRE.

5.2.1 Experimental setup

MLP-UNI is trained on 63 h of data from five European languages, namely British English, Italian, Spanish, Swiss French and Swiss German (see also Section 2.5, page 22). Since all the SpeechDat(II) dictionaries use SAMPA symbols, we merge phones that share the same SAMPA symbol across languages to build the universal phone set that contains 117 phones. A summary of the universal MLP training is shown in Table 3.1, page 38.

For the non-native ASR experiments, we use the HIWIRE dataset. Usually, the first 50 utterances of each speaker serve as adaptation data. As shown in Table 5.1, we further reduce the amount of adaptation data by using fewer utterances per speaker. The 3-minutes and the 2-minutes datasets are obtained by the same heuristic as already described in Section 3.3.5, page 44.

The tied states KL-HMM system, as described in Chapter 4, uses the KL divergence based decision tree algorithm, which requires to tune the minimum occupancy threshold and the minimum decrease in the cost function threshold. As already done for the study on Greek data in Chapter 4, we fix the the minimum occupancy threshold to 20. Since the HIWIRE dataset does not provide a development set to tune thresholds, we take the first 30 utterances of each speaker as training data (90 min). We then use utterances 31-50 of each speaker as development set to tune the word insertion penalty, and the minimum decrease in the cost function threshold. For all our subsequent experiments, we then use the same tuned parameters.

Table 5.1: Utterance choice on the HIWIRE dataset to simulate low amount of data.

Amount of data [min]	Considered Utterances
149	Utterances 1-50
90	Utterances 1-30
32	Utterances 1-10
16	Utterances 5-9
10	Utterances 3,5,7
3	Manually selected
2	Manually selected

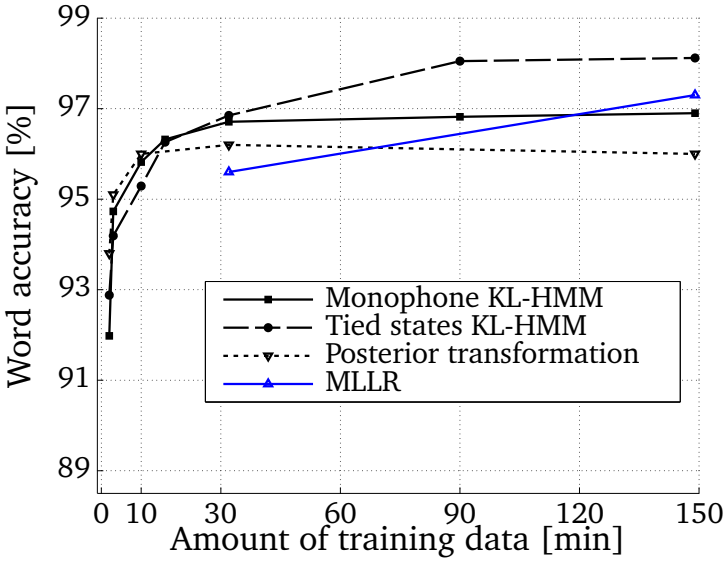


Figure 5.1: Word accuracies on the spoken pilot order recognition task of the HIWIRE database, with 133 different words and a grammar perplexity of 14.9, for different amounts of training data. Multilingual KL-HMM using state tying and multilingual monophone KL-HMM are applied to non-native ASR. The performance of the multilingual posterior transformation as reported in Table 3.6, page 44, is also shown. Furthermore, as an additional reference point, speaker-dependent MLLR results, as reported by Segura et al. [2007], are also given. It is however important to keep in mind that the MLLR results come from a different implementation.

5.2.2 Results

Figure 5.1 shows the resulting performance in terms of word accuracy as a function of amount of training data on the spoken pilot order task of the non-native English database, HIWIRE. There are 133 different words and the grammar perplexity is 14.9 [Segura et al., 2007].

As expected, the tied states KL-MM outperforms the monophone KL-HMM (presented in Section 4.4) due to the increased number of states in the target HMM. For comparison, Figure 5.1 also shows the performance of the multilingual posterior transformation as reported in Table 3.6, page 44. As an additional reference point, speaker-dependent MLLR results, as reported by Segura et al. [2007], are also given in the figure. It is however important to keep in mind that they come from a different implementation. A detailed comparison of the tied states KL-HMM system to related approaches is given later in Section 5.4.

5.3 Crosslingual KL-HMM

We have already seen that a multilingual MLP can significantly improve the ASR performance of non-native ASR. In this section, we investigate crosslingual KL-HMMs, i.e. the MLP is trained on a different language than the categorical distributions. Section 5.3.1 presents the experimental setup and Section 5.3.2 the results.

Table 5.2: Overview over four MLPs used to estimate posterior features. The total amount of training data, the frame accuracy on the development data (DEV acc.), as well as the phoneme set including the number of phonemes (S) are given.

System	Phoneme set	Number of phonemes (S)	TRN data	DEV acc.
MLP-ES	SAMPA Spanish	32	11.5 h	73.2%
MLP-IT	SAMPA Italian	52	11.5 h	68.6%
MLP-SF	SAMPA French	42	13.5 h	65.5%
MLP-SZ	SAMPA German	59	14.1 h	60.4%

5.3.1 Experimental setup

We estimate the posteriors features with four different MLPs trained on data from SpeechDat(II). We train one MLP for each of the following European languages: Spanish (MLP ES), Italian (MLP IT), Swiss French (MLP SF) and Swiss German (MLP SZ), respectively. Similarly to the earlier experiments performed with SpeechDat(II) data, we only use *Corpus S*, which contains ten read sentences per speaker and we split the databases into training (1,500 speakers), development (150 speakers) and test (350 speakers) sets as described in Section 2.5, page 23. For the MLP training, we only use the training and development sets.

All the MLPs are trained from 39 MF-PLP features in a nine frame temporal context as input. As we usually do, we fix the number of parameters in each MLP to 10% of the number of available training frames. Table 5.2 gives an overview over the four MLPs including the number of outputs (number of phonemes S), the amount of training data and the frame accuracies on the development data.

For the crosslingual studies, we explore monophone and tied states KL-HMM systems that use three states per phone. We always use all the adaptation data for the training of the categorical distributions. For the tuning of the word insertion penalty, we also use the adaptation data. For state tying, we use the same thresholds as in Section 5.2. Evaluation is performed on the test set.

5.3.2 Results

Recall that HIWIRE is a non-native English speech corpus that contains English utterances pronounced by natives of France (31 speakers), Greece (20 speakers), Italy (20 speakers) and Spain (10 speakers). Intuitively, we hypothesize MLP-SF to perform better on French accented data, MLP-ES to perform better on Spanish accents and so on. Additionally, for comparison, we do not train a system on Greek data (to keep one unseen non-native accent data set for testing). However, we train a system on Swiss German (MLP-SZ), a non-native accent that is not present in the HIWIRE data.

We evaluate monophone as well as tied states KL-HMM systems in a crosslingual setup. Results for monophone KL-HMM systems are presented in Table 5.3. The last column (TST) shows

Table 5.3: Word accuracies on the spoken pilot order task of the HIWIRE database with133 different words and a grammar perplexity of 14.9. Comparison of monophone KL-HMM systems where the MLP is trained on SpeechDat(II) data from different languages (see Table 5.2). FR stands for French-, GR for Greek-, IT for Italian- and SP for Spanish-accented English, respectively. Best results of each column are marked bold; italic numbers point to results that are not significantly worse.

System	FR	GR	IT	SP	TST
MLP-ES	92.6%	95.1%	92.4%	**93.6%**	93.3%
MLP-IT	*93.6%*	**96.1%**	**93.9%**	*93.4%*	**94.2%**
MLP-SF	**93.8%**	92.7%	91.7%	92.1%	92.8%
MLP-SZ	*93.6%*	*95.2%*	92.4%	*92.9%*	93.6%

the performance on the whole test set. The other columns report the performance on the data of speakers with a particular accent (note that the acronyms for Spanish differ because SpeechDat(II) officially uses *ES* and HIWIRE *SP*). The best result of each column is marked bold. Italic numbers point to results that are not significantly worse than the best result. Recall that we use the bootstrap estimation method [Bisani and Ney, 2004] and a confidence interval of 95%, as described in Section 2.4, for all the significance tests.

As expected, MLP-SF performs best on French non-native speech, MLP-IT performs best on Italian non-native speech and MLP-ES performs best on Spanish non-native speech. The Swiss German models do not perform best on any of the accents.

System MLP-IT has the best average performance but, as hypothesized, the performance is significantly worse compared to system MLP-EN (95.0%). Interestingly, Raab et al. [2008] also evaluated native German, Italian, Spanish and French models on HIWIRE data. The performance they reported is lower than what we report here, but Italian still outperformed all other models.

The results of the tied states KL-HMM experiments are shown in Table 5.4. There is no signifi-

Table 5.4: Word accuracies on the spoken pilot order recognition task of the HIWIRE database with133 different words and a grammar perplexity of 14.9. Comparison of tied states KL-HMM systems where the MLP is trained on SpeechDat(II) data from different languages (see Table 5.2). FR stands for French-, GR for Greek-, IT for Italian- and SP for Spanish-accented English, respectively. Best results of each column are marked bold; italic numbers point to results that are not significantly worse.

System	FR	GR	IT	SP	TST
MLP-ES	96.3%	*97.4%*	*95.9%*	**96.4%**	**96.5%**
MLP-IT	96.3%	**97.5%**	**96.1%**	*95.9%*	**96.5%**
MLP-SF	*96.7%*	96.8%	*95.5%*	95.2%	*96.2%*
MLP-SZ	**97.0%**	*97.4%*	95.2%	*96.2%*	**96.5%**

cant difference between the results on the complete test set for all four different languages (column *TST*). However, they all perform significantly worse than the English system (97.2 %). If the individual accents are analyzed separately, there are significant differences between the systems. For the Spanish- and the Italian-accented speech, the system that uses the MLP trained on the corresponding language performs best as expected. For the French-accented speech, the system based on the Swiss German MLP performs marginally (not statistically significant) better than the system based on the Swiss French MLP.

5.4 Comparison with related work

In this section, we discuss the relationship between KL-HMM and PAM, LHN, MLLR, SCHMM, and ML-tag systems. We start the section with a discussion of the KL-HMM local score given in (3.14), page 35, and recalled here:

$$d(\boldsymbol{P}_t, \boldsymbol{Q}^\ell) = \sum_{k=1}^{S} P_{t,k} \log\left(\frac{P_{t,k}}{Q_k^\ell}\right), \tag{5.1}$$

where $\boldsymbol{P}_t$ is the observed feature vector at time t, and $\boldsymbol{Q}^\ell$ the reference vector associated with HMM state ℓ.

5.4.1 Semi-continuous HMM (SCHMM)

As already seen earlier, in Chapter 4, the KL divergence is not symmetric and Aradilla [2008] studied different variants of KL divergence based local scores for the KL-HMM framework. Given the posterior feature at time t, $\boldsymbol{P}_t$, and the HMM state distribution of state ℓ, $\boldsymbol{Q}^\ell$, the following local scores have been introduced:

$$d_{KL} = d(\boldsymbol{Q}^\ell, \boldsymbol{P}_t) = \sum_{k=1}^{S} Q_k^\ell \log\left(\frac{Q_k^\ell}{P_{t,k}}\right), \tag{5.2}$$

$$d_{RKL} = d(\boldsymbol{P}_t, \boldsymbol{Q}^\ell) = \sum_{k=1}^{S} P_{t,k} \log\left(\frac{P_{t,k}}{Q_k^\ell}\right), \tag{5.3}$$

$$d_{SKL} = \frac{1}{2} d_{KL} + \frac{1}{2} d_{RKL}. \tag{5.4}$$

In this thesis, we always use d_{RKL} for KL-HMM training and decoding and d_{KL} for the decision tree clustering algorithm. Different local scores result in different estimates for Q_k^ℓ [Aradilla, 2008]:

$$Q_k^\ell = \frac{1}{Z}\left(\prod_{t=1}^{T^\ell} P_{t,k}\right)^{\frac{1}{T^\ell}} \quad \text{for } d_{KL} \text{ (normalized geometric mean)}, \tag{5.5}$$

$$Q_k^\ell = \frac{1}{T^\ell}\sum_{t=1}^{T^\ell} P_{t,k} \quad \text{for } d_{RKL} \text{ (arithmetic mean)}, \tag{5.6}$$

where Z acts as a normalization constant. For d_{SKL}, there is no closed form solution.

The standard Viterbi algorithm maximizes the likelihood $p(\boldsymbol{x}_t|\Omega)$. In SCHMM [Huang and Jack, 1989], recalling (2.17), page 16, each state d^ℓ is parametrized as:

$$p(\boldsymbol{x}_t|\Omega, d^\ell) = \sum_{k=1}^{S} c_k^\ell p_k(\boldsymbol{x}_t|\Omega_k), \tag{5.7}$$

where the probability density function p_k of s^k, the k^{th} Gaussian distribution, is parametrized with $\Omega_k = \{\boldsymbol{\mu}_k, \Sigma_k\}$, shared among all the states, and the weights c_k^ℓ are estimated for each state individually.

In SCHMM, we assume that Ω_k is fixed $\forall k$ and only c_k^ℓ needs to be estimated. The well-known maximum likelihood solution for c_k^ℓ (see, e.g., Bilmes [1998]) is given by:

$$c_k^\ell = \frac{1}{T^\ell} \sum_{t=1}^{T^\ell} p(s^k|\boldsymbol{x}_t, \Omega_k), \tag{5.8}$$

where the sum extends over all t associated to a state d^ℓ. In that particular context, it follows from the close resemblance of (5.6) and (5.8) that estimating c_k^ℓ along a maximum likelihood criterion is equivalent to estimating Q_k^ℓ if d_{RKL} is used.

5.4.2 Probabilistic acoustic mapping (PAM)

PAM, introduced by Sim [2009, Section IV.C], estimates the target phone probability $P(d_t^\ell|\boldsymbol{x}_t)$ as follows:

$$P(d_t^\ell|\boldsymbol{x}_t) = \frac{1}{Z} \exp\Big(\sum_{k=1}^{S} W_{\ell,k} \log P(s_t^k|\boldsymbol{x}_t) + b_\ell \Big), \tag{5.9}$$

where Z acts as a normalization factor. $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weight matrix and the bias vector of an MLP, respectively. Recall, the relative entropy between two discrete random variables of dimensionality S:

$$H(\boldsymbol{P}, \boldsymbol{Q}) = -\sum_{k=1}^{S} P_k \log Q_k. \tag{5.10}$$

We can rewrite (5.9) as:

$$P(d_t^\ell|\boldsymbol{x}_t) = \frac{\exp\Big(- H(\boldsymbol{W}^\ell, \boldsymbol{P}_t) + b_\ell \Big)}{\sum_{j=1}^{D} \exp\Big(- H(\boldsymbol{W}^j, \boldsymbol{P}_t) + b_j \Big)}, \tag{5.11}$$

where $\boldsymbol{W}^\ell$ are the weights associated with the ℓ^{th} MLP output. If the MLP is trained with the cross-entropy criterion (see (2.4), page 12, for the cross-entropy based MLP training) the local

Table 5.5: Word accuracies on the test data of the HIWIRE data set. For all the experiments all the adaptation data was used for training. Linear PAM consists of a two-layer MLP and non-linear PAM of a three-layer MLP as described in [Sim, 2009]. RKL-tied corresponds to the multilingual KL-HMM system using state tying, presented in Section 5.2, and performs significantly better than all other systems.

System	Score	Re-align	Linear	Context	Word accuracy
KL-mono	d_{KL}	embedded	yes	no	96.7%
KL-tied	d_{KL}	embedded	yes	yes	97.6%
PAM	d_{PAM}	no	yes	no	96.2%
PAM	d_{PAM}	yes	yes	no	96.9%
PAM	d_{PAM}	no	no	no	97.1%
PAM	d_{PAM}	yes	no	no	97.4%
RKL-tied	d_{RKL}	embedded	yes	yes	**98.1%**

score d_{PAM} that is minimized can be written as:

$$d_{PAM} = -\log P(d_t^\ell | \boldsymbol{x}_t) \propto \Big(H(\boldsymbol{W}^\ell, \boldsymbol{P}_t) - b_\ell \Big). \tag{5.12}$$

Using (5.10), Equations (5.3) and (5.2) can be rewritten in terms of the entropy as:

$$d_{RKL} = H(\boldsymbol{P}_t, \boldsymbol{Q}^\ell) - H(\boldsymbol{P}_t, \boldsymbol{P}_t), \tag{5.13}$$

$$d_{KL} = H(\boldsymbol{Q}^\ell, \boldsymbol{P}_t) - H(\boldsymbol{Q}^\ell, \boldsymbol{Q}^\ell). \tag{5.14}$$

Hence, d_{KL} and d_{PAM} are closely related and $H(\boldsymbol{Q}^\ell, \boldsymbol{Q}^\ell)$ in d_{KL} acts as a target dependent bias. For d_{RKL} however, the bias is source dependent: $H(\boldsymbol{P}_t, \boldsymbol{P}_t)$.

In the following, we summarize the differences between KL-HMM, with d_{RKL} as local score, and PAM. Table 5.5 shows how these differences affect the WACC.

- Cost function: d_{RKL} performs better than d_{KL}, which performs similar to d_{PAM}.
- Embedded re-alignment: both, PAM and the proposed approach allow to benefit from re-alignment. In the case of PAM, a re-alignment requires the MLP to be retrained. As seen in Table 5.5, PAM with re-alignment yields a better performance than PAM without re-alignment.
- Context-dependent models: in theory, both approaches can benefit from context-dependent models. In practice however, due to data sparsity, usually state tying is required. We developed an algorithm to perform state tying at the KL-HMM state level. In the case of PAM, it is not obvious how to tie MLP outputs to train a context-dependent recognizer on limited amounts of data.

Note that the optimal number of hidden units for the non-linear PAM approach was 800-900 in [Sim, 2009]. To evaluate wether more hidden units yield a better performance, we doubled

Table 5.6: Comparison of word accuracies on the test data of the HIWIRE data set. As an additional reference point, we show the LHN results reported in [Gemello et al., 2007]. However, the results are only conditionally comparable since the KL-HMM systems (RKL-mono and RKL-tied) are trained on 8kHz multilingual data, and the LHN systems on 16 kHz English data.

System	Adaptation	MLP trained on	Word accuracy
LHN	Speaker-based	English 16 kHz	95.4 %
LHN	Data-based	English 16 kHz	98.2 %
RKL-mono	Speaker-based	Multilingual 8 kHz	96.1 %
RKL-tied	Data-based	Multilingual 8 kHz	98.1 %

the amount of hidden units and found a marginal improvement. Therefore, we report the performance of the latter configuration in Table 5.5. We also investigated more than one re-alignment iteration for PAM, but did not observe further improvement.

5.4.3 Linear hidden network (LHN)

The linear hidden network (LHN) is another MLP-based adaptation approach to perform a hidden feature transformation [Gemello et al., 2007]. The LHN is applied to the activations of the internal layer and can be trained using the standard back-propagation algorithm while keeping frozen the weights of the original network. Once the LHN is trained, it is combined with the original (unadapted) weights:

$$\boldsymbol{W}_a = \boldsymbol{W}_{LHN} \times \boldsymbol{W}_{ORIG}$$
$$\boldsymbol{b}_a = \boldsymbol{b}_{LHN} \times \boldsymbol{W}_{ORIG} + \boldsymbol{b}_{ORIG}$$

where $\boldsymbol{W}_a$ and $\boldsymbol{b}_a$ are the weights and the bias of the adapted layer, $\boldsymbol{W}_{ORIG}$ and $\boldsymbol{b}_{ORIG}$ are the weight and bias of the layer following the LHN in the original unadapted network, and $\boldsymbol{W}_{LHN}$ and $\boldsymbol{b}_{LHN}$ are the weight and the biases of the LHN.

KL-HMM differs from LHN in many aspects, including the ones already listed at the end of Section 5.4.2. Additionally, LHN is bound to a given and fixed phoneme set and is therefore closely related to MLLR. Based on hidden layer adaptation, it is not obvious how to apply phone space transformations. To use an already trained *original* MLP, it needs to be trained from aligned data that makes use of the same phoneme set (targets) than the adaptation data.

Gemello et al. [2007] used LHN to adapt an MLP, previously trained on native English, to the HIWIRE data. They investigated speaker-based adaptation (one LHN per speaker) and data-based adaptation (one LHN for all data). As shown in Table 5.6, the data-based LHN results in similar performance than the tied states KL-HMM system presented in Section 5.2, system RKL-tied. For the speaker-based LHN adaptation, they adapted and tested for each speaker separately. Not every speaker pronounced each phone in the first 50 utterances (adaptation set). Therefore, we investigate a monophone KL-HMM (RKL-mono) instead of a tied states

Table 5.7: Word accuracies on the test data of the HIWIRE data set. For all the experiments all the adaptation data was used for training. Results on TIMIT were reported in [Segura et al., 2007].

System	Seed trained on	kHz	Word accuracy
MLLR	TIMIT	16	97.3%
MLLR	SpeechDat(II) English	8	95.7%
MLLR	SpeechDat(II) multilingual	8	95.7%
RKL-tied	SpeechDat(II) English	8	97.2%
RKL-tied	SpeechDat(II) multilingual	8	98.1%

KL-HMM on a per-speaker basis. For the phones without adaptation data, the categorical distributions are never updated and keep the initial values. RKL-mono outperforms the speaker-based LHN. Note that the results in Table 5.6 are only given as a reference point since the proposed approach was trained on 8 kHz multilingual data, and LHN on 16 kHz English data. Gemello et al. [2007] also performed decoding with unadapted acoustic models and reported significantly lower recognition accuracies if 8 kHz data was used during training. Therefore, they only adapted models trained on 16 kHz data.

5.4.4 Maximum likelihood linear regression (MLLR)

MLLR has been widely used to perform acoustic model adaptation for HMM/GMM based recognizers. Segura et al. [2007] also applied conventional MLLR speaker adaptation with HTK to adapt models trained on TIMIT, a well-known acoustic-phonetic continuous speech corpus, to the HIWIRE database. To give another reference point, we apply the manual mappings, given in Table A.2, page 103, and perform speaker-based MLLR with HTK to adapt the SpeechDat(II) English and multilingual seed models to HIWIRE.

It can be seen in Table 5.7 that the multilingual data does not improve the word accuracy on HIWIRE if MLLR is used. We attribute the performance difference between MLLR on TIMIT and SpeechDat(II) English to the different nature of the data such as sampling frequency, microphone, and background noise.

5.4.5 Language-independent acoustic models (ML-tag)

Furthermore, we can compare our work to the estimation of language-independent acoustic models using the ML-tag method [Schultz and Waibel, 2001], also introduced in Chapter 2. Recalling (2.18), page 17:

$$p(\boldsymbol{x}_t|\Omega, d^\ell) = \sum_{n=1}^{N} c_n^\ell \mathcal{N}(\boldsymbol{x}_t|\Omega_n^\ell), \tag{5.15}$$

where N are the number of Gaussians used to model state d^ℓ. HMM states across different languages share the Gaussian components Ω_n^ℓ if they are represented with the same IPA symbol. The mixture weights c_n^ℓ however, are trained for each HMM state individually.

Hence, ML-tag uses a pool of N Gaussians for each universal phone. In that case, language specific phone models are then obtained by estimating language dependent weights, acting as similarity measure between universal and monolingual phones.

The proposed KL-HMM system can convert universal phone posteriors to any language. Thus, it can be seen as a discriminative approach of estimating language-independent acoustic models with the ML-tag method.

5.5 Conclusion

In this chapter, we have evaluated KL-HMM systems in the specific context of accented speech recognition, involving high phone acoustic variability and phone set mismatches between (multilingual) phone sets. KL-HMM training iteratively optimizes a principled KL divergence based function, which was shown to be amenable to posterior distributions.

The resulting system has been shown to be able to efficiently exploit multi- and crosslingual adaptation data, using a parsimonious number of parameters while also being particularly well suited in the case of phone set mismatch. This conclusion is further supported by additional evidence and theoretical and experimental comparisons with similar approaches such as PAM, LHN and MLLR.

In the next chapter, we investigate how KL-HMM can improve ASR for under-resourced languages.

6 Under-resourced ASR

We have shown earlier, in Chapter 4, that ASR may benefit from data in languages other than the target language, especially in the case when there is less than one hour of training data for the language to be recognized. However, in Chapter 4, we only simulated an under-resourced language by artificially reducing the amount of available training data.

In this chapter, taking Afrikaans as a representative of a *real* under-resourced language, we report how to boost the performance of an under-resourced Afrikaans ASR system by using already available Dutch data.

We use three different acoustic modeling techniques, namely KL-HMM, Tandem as well as subspace Gaussian mixture models (SGMMs) to optimally exploit available multilingual resources. In the case of KL-HMM and Tandem, this is done through posterior features estimated by an MLP, and in the case of SGMMs, this is done through parameter sharing. We show that all three resulting multilingual systems yield improvement compared to a conventional monolingual HMM/GMM system only trained on Afrikaans. Furthermore, we show that KL-HMMs are extremely powerful for under-resourced languages: using only six minutes of Afrikaans data (in combination with out-of-language data), KL-HMM yields about 30% relative improvement compared to conventional MLLR and maximum a posteriori (MAP) based acoustic model adaptation.

6.1 Related work

Developing a state-of-the-art speech recognizer from scratch for a given language is expensive. The main reason for this is the large amount of data that is usually needed to train current recognizers. Data collection involves large amounts of manual work, not only in time for the speakers to be recorded, but also for annotation of the subsequent recordings. Therefore, the need for training data is one of the main barriers in porting current systems to many languages. On the other hand, large databases already exist for many languages.

Engelbrecht and Schultz [2005] used multilingual seed models to bootstrap an Afrikaans speech recognizer. To adapt the multilingual models to the Afrikaans language, they simply retrained the models with 1,000 Afrikaans utterances. The adapted models performed more than 50 % relative better than the unadapted models. On the other hand, Niesler [2007] studied the sharing of resources on under-resourced languages, including Afrikaans, inspired by multilingual acoustic modeling techniques proposed by Schultz and Waibel [2001]. However, only marginal ASR performance gains were reported. We found in the previous chapters that the relation between phonemes of different languages can be learned and exploited for crosslingual acoustic model training or adaptation. Furthermore, we found that posterior features, estimated by multilayer perceptrons (MLPs), are particularly well suited for such tasks. Previous posterior feature studies that used more than one hour of target language data reported rather small or no improvements (up to 5% relative) [Stolcke et al., 2006, Tòth et al., 2008, Grézl et al., 2011]. However, those studies sometimes focused on languages that are very different such as English and Mandarin [Stolcke et al., 2006] or English and Hungarian [Tòth et al., 2008].

Vu et al. [2010] also studied scenarios where no transcriptions are available for the under-resourced target language data. In this study, we assume to have transcriptions for the under-resourced language data. However, we will show that we can limit the amount of needed, transcribed data to a minimum.

In one of our initial studies [Imseng et al., 2012c], we focused on an Afrikaans ASR system and used posterior features estimated by MLPs that were trained on similar languages such as English, Dutch and Swiss German. We compared Tandem and KL-HMM, which are both able to exploit multilingual information in the form of posterior features and investigated the following aspects:

- *Crosslinguality:* We studied how out-of-language data can be used to improve ASR performance of an under-resourced language. Due to the lack of an appropriate Afrikaans language model, we reported phoneme accuracies. More specifically, in the KL-HMM and Tandem setup, we explored systems, where the MLP is trained on data from a language different from the target language. We also briefly discussed if there is a relation between similarity of the other language and performance gain on the target language.

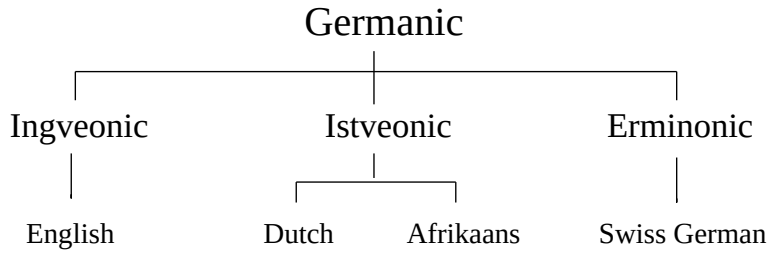


Figure 6.1: Afrikaans in the context of the other considered Germanic languages: Dutch, English and Swiss German.

Table 6.1: Summary of the initial study on boosting Afrikaans ASR with out-of-language data. Relative increase in phoneme accuracy compared to a monolingual system only trained on Afrikaans is shown. KL-HMM gains more from multilinguality and Tandem from context-dependency. In both cases, the gains are additive. The baseline results are phoneme accuracies.

Relative gain with	KL-HMM	Tandem
monolingual (baseline)	58.7 %	61.2 %
+(Dutch) context	+2.6 %	+8.7 %
+bilingual (Afrikaans-Dutch) context	+10.6 %	+10.3 %
+multilingual context	+11.4 %	+10.8 %

According to the tree in [Blažek, 2005], and also shown in Figure 6.1, Afrikaans and Dutch are Istveonic Germanic languages whereas British English and Swiss German are also Germanic languages, but located on different branches, namely Ingveonic and Erminonic Germanic, respectively. Intuitively, we would expect that Dutch data should provide most benefit. A similarity analysis of Heeringa and de Wet [2008] underpins this assumption and our studies confirmed it. Indeed, the crosslingual setup with the Dutch MLP outperformed the systems that used the MLPs trained on Swiss German or English data.

- *Context-dependency:* Since there is a relatively large amount of Dutch data available, we enriched the exploited out-of-language information by adding context dependency, i.e. we trained the MLPs on context-dependent targets. As shown in Table 6.1, where we report relative increase in phoneme accuracy compared to a monolingual KL-HMM/Tandem system only trained on Afrikaans data, there is more improvement for the Tandem systems.

- *Multilingual context-dependency:* We combined the resources of multiple languages in the form of posterior features by concatenating MLP outputs. The MLPs were all trained on context-dependent targets. In Table 6.1, we distinguish between *+bilingual context* and *+multilingual context*. In the first case, we only combined the MLP outputs of an Afrikaans and a Dutch MLP whereas in the latter case, we combined the MLP outputs of all four involved languages. Table 6.1 reveals that given the output of an MLP trained on Dutch, there is only marginal improvement if the output of MLPs trained on English and Swiss German data are used as well.

Hence, for the subsequent experiments reported in this chapter, we will only use Dutch data to boost the performance of an Afrikaans ASR system. Furthermore, in [Imseng et al., 2012c], we limited ourselves to MLPs with relatively small numbers of context-dependent targets (about 200). Here, we investigate MLPs trained on context-dependent targets with ten times more output units and compare the aforementioned acoustic modeling techniques to SGMM and conventional adaptation techniques such as MLLR and MAP adaptation. Note that we used the symmetric version of the KL divergence, d_{SKL}, for the experiments in [Imseng et al., 2012c].

Based on the findings of the last chapter, here, we always employ d_{RKL} for the HMM training and decoding.

6.2 Data

We use data from Afrikaans and Dutch as summarized in Table 6.2. The two databases are described in Section 2.5, page 24. Here, we only give a brief overview.

6.2.1 Afrikaans

The Afrikaans data is available from the Lwazi corpus [Barnard et al., 2009] that consists of utterances pronounced by 200 speakers, recorded over a telephone channel at 8 kHz. Each speaker produced approximately 30 utterances, where 16 were randomly selected from a phonetically balanced corpus and the remainder consisted of short words and phrases.

The dictionary [Davel and Martirosian, 2009] that we use contains 1,585 different words and makes use of a phoneme set containing 38 phonemes, also shown in Table A.1, page 102. The training and test sets are provided by the HLT group at Meraka. In total, about three hours of training data and 50 minutes of test data is available (after voice activity detection).

Since we do not have access to an appropriate language model, we train a bi-gram phoneme model on the training set and only report phoneme accuracies. The bi-gram phoneme model can learn the phonotactic constraints of the Afrikaans language and has a phoneme perplexity of 14.5 on the training set and 14.7 on the test set.

6.2.2 Dutch

We use data of the Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN) [Oostdijk, 2000] that contains standard Dutch pronounced by more than 4,000 speakers from the Netherlands and Flanders. We only use *Corpus o* because it contains phonetically aligned *read* speech data pronounced by 324 speakers from the Netherlands and 150 speakers from Flanders. *Corpus o* uses 47 phonemes, also shown in Table A.1, page 102, and contains 81 h of data after the deletion of silence segments that are longer than one second. It was recorded at 16 kHz, but since we use the data to perform ASR on Afrikaans, we downsample it to 8 kHz prior to feature extraction.

6.3 Multilingual boosting strategies

The investigated approaches are well suited to exploit out-of-language data. Two of the presented approaches exploit out-of-language data on the feature level using posteriors, namely Tandem and KL-HMM. The posterior feature based approaches exploit out-of-language infor-

Table 6.2: Overview over MLPs trained on Dutch and Afrikaans data. The number of output units, the amount of training data and the frame accuracy on the cross-validation set is given.

ID	Language	Number of phonemes	Number of tied states	Amount of training data	Frame accuracy on validation set
AF	Afrikaans	38	187	3 h	43.8%
CGN	Dutch	47	1,789	81 h	56.5%

mation in the form of an MLP which is trained on out-of-language data. SGMMs on the other hand exploit out-of-language data on the acoustic model level and use a universal background model (UBM) and shared projection matrices trained on out-of-language data.

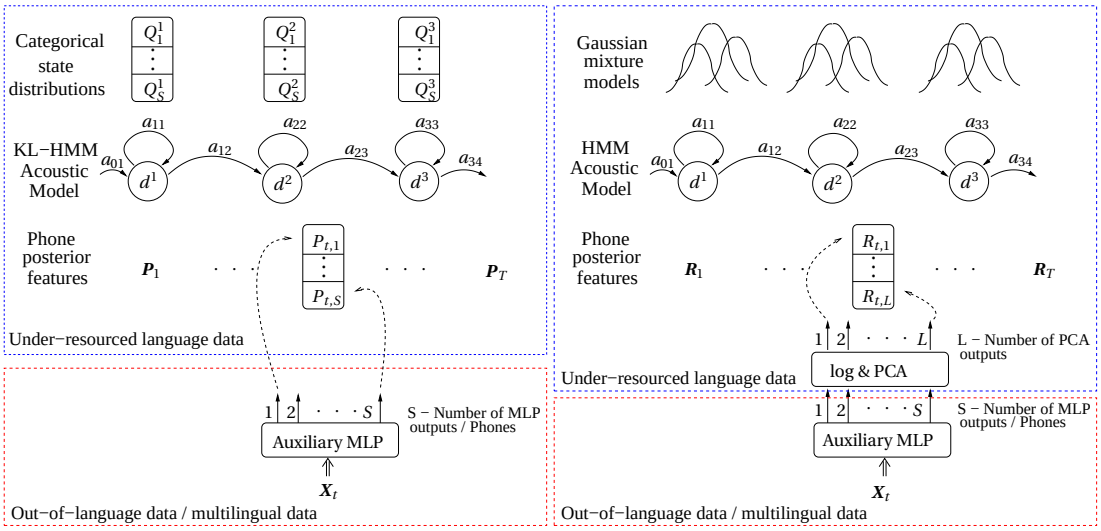
6.3.1 Feature level approach

For each language (Afrikaans and Dutch), as usual, we use 39 MF-PLP features as input to the MLP. In our earlier study [Imseng et al., 2012c], we found that systems that use MLPs which are trained on context-dependent targets (triphones) outperform MLPs trained on context-independent monophones. Therefore, we train both MLPs on triphone targets. To obtain triphone targets, we develop a standard HMM/GMM system with all the training data for both languages independently and use the standard likelihood based decision tree approach to tie rare triphones. More specifically, we use the MDL criterion to automatically determine the number of tied triphones for each language independently [Shinoda and Watanabe, 1997]. As described by Shinoda and Watanabe [1997], the MDL criterion has a hyper-parameter, c, which controls the weight of the term that penalizes models with large amounts of triphones. We tune c on the Afrikaans database and fix it to 16 (for both databases). The HMM/GMM systems are then used to align the training data in terms of tied states.

During MLP training, the tied states alignment is used to obtain labels and we use 90% of the training set for training and 10% for cross-validation to stop training. Table 6.2 shows the number of output units (tied states) per MLP and also the frame accuracy on the cross-validation set.

The HMM states d^{ℓ}, with $\ell \in \{1, \cdots, D\}$, are associated with the target language. Each triphone of the target language is modeled with three states and D is the total number of tied states. For both KL-HMM and Tandem, we fix the transition probabilities to 0.5, as already discussed at the end of Section 4.1, page 49. The emission probabilities are trained from within-language data only. Here, we assume that we have access to a limited amount of within-language data and vary the amount of Afrikaans data from six minutes to three hours.

As described in Chapter 4, and also shown in Figure 6.2, KL-HMM uses a categorical distribution to model the posteriors features. As shown in Figure 6.2, Tandem uses GMMs. Therefore, the posterior features $\boldsymbol{P}_t$ need to be post-processed. More specifically, the log-phone posteriors are decorrelated with a principal component analysis (PCA). The transfor-



(a) KL-HMM (b) Tandem

Figure 6.2: Illustrative comparison of the two feature-level based approaches, KL-HMM and Tandem. KL-HMM directly models the raw posteriors using categorical distributions. Tandem uses GMMs and therefore models decorrelated log-phone posteriors. GMMs and categorical distributions are trained on target language data and the MLP can be trained on out-of-language data.

mation matrix can be estimated on within-language data. Usually, the resulting feature vector $\boldsymbol{R}_t = [R_{t,1}, \cdots, R_{t,L}]^\top$, has a reduced dimensionality L.

6.3.2 Acoustic model level approach

To exploit out-of-language data, the SGMM model parameters can be divided into HMM-state specific and shared parameters, as visualized in Figure 6.3 and also given hereafter:

$$p(\boldsymbol{x}_t|\Omega, d^\ell) = \sum_{i=1}^{I} c_i^\ell \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_i^\ell, \Sigma_i), \tag{6.1}$$

$$\boldsymbol{\mu}_i^\ell = \mathbf{M}_i \boldsymbol{v}^\ell, \tag{6.2}$$

$$c_i^\ell = \frac{\exp(\boldsymbol{w}_i \cdot \boldsymbol{v}^\ell)}{\sum_{j=1}^{I} \exp(\boldsymbol{w}_j \cdot \boldsymbol{v}^\ell)}, \tag{6.3}$$

where all the states share the same I Gaussians. The model in each HMM state is then represented by a simple GMM with I Gaussians, mixture weights c_i^ℓ, means $\boldsymbol{\mu}_i^\ell$, and covariances Σ_i. The latter are shared across all states. The state-specific vectors $\boldsymbol{v}^\ell \in \mathbb{R}^U$ together with the globally shared parameters $\boldsymbol{M} = [\boldsymbol{M}_1, \cdots, \boldsymbol{M}_I]^\top$, where each $\boldsymbol{M}_i$ is a $C \times U$ matrix with C being the dimensionality of the cepstral features, and $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_I]^\top$ with $\boldsymbol{w}_i = [w_i^1, \cdots, w_i^U]$ are

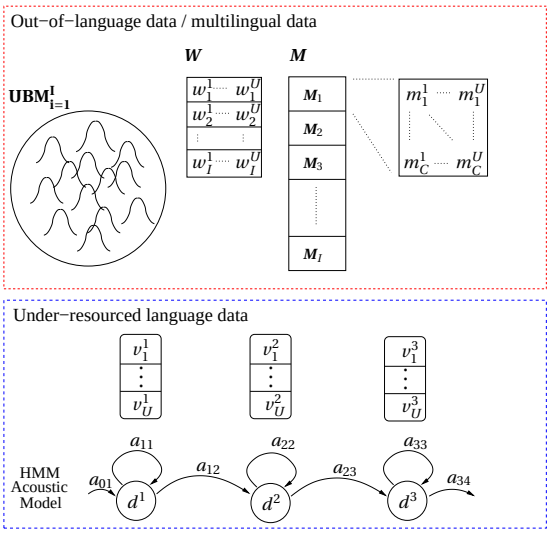


Figure 6.3: Out-of-language data exploitation with SGMMs. HMM-state specific parameters are trained on the target language and the shared parameters can be trained on out-of-language data.

used to derive the means and mixture weights representing the given HMM state. For the initialization of the SGMM, a generic GMM with I Gaussians, denoted as UBM, modeling all the speech is used.

As proposed by Burget et al. [2010], the projection matrices $\boldsymbol{M}$ and $\boldsymbol{W}$ together with the UBM can be perceived as shared (language-independent) and can therefore be trained using large amounts of data from different languages.

Equations (6.1), (6.2) and (6.3) assume (without loss of generality) one state-specific vector $\boldsymbol{v}^{\ell}$ to be assigned to each HMM state. However, we model each state with a mixture of sub-states [Povey et al., 2011], each having its own sub-state specific vector $\boldsymbol{v}_j^{\ell}$, where $j = 1, \cdots, J_d$ with J_d being the number of sub-states of state d. In that case, we also need to estimate the mixture weights c_j for each sub-state. The sub-state-specific vectors $\boldsymbol{v}_j^{\ell}$ as well as the weights c_j are trained on within-language data.

6.4 Systems

In this section, we describe the systems that we investigate to study the exploitation of out-of-language data in the framework of under-resourced ASR. Scaling factor and phoneme insertion penalty are for each system individually tuned on the cross-validation set.

Table 6.3: The Afrikaans phonemes without a matching Dutch seed model (same IPA symbol not present in the Dutch phoneme set) are given in the left column. The corresponding manually chosen Dutch seed models are listed in the right column.

Afrikaans	Dutch
ɑː	ɑ
ae	ɛ
oe	ʏ
øː	ø
ɦ	h

6.4.1 HMM/GMM

Each context-dependent triphone is modeled with three states. As usually done, we first train context-independent monophone models, which serve as seed models for the context-dependent triphone models. We use a mixture of eight Gaussians per state to model the emission probabilities. To balance the number of parameters with the amount of available training data, we apply conventional state tying with a decision tree that is based on the MDL principle [Shinoda and Watanabe, 1997].

6.4.2 Maximum likelihood linear regression (MLLR)

To evaluate whether an under-resourced language can be accommodated by linear transforms, we first train a triphone HMM/GMM system on the Dutch data. Each triphone state is modeled with a mixture of 16 Gaussians. We then investigate the standard MLLR and use a regression tree that allows up to 32 regression classes.

For most Afrikaans phonemes, we use the corresponding Dutch phoneme, represented with the same IPA symbol, as a seed model for MLLR. However, not all the Afrikaans phonemes are present in the Dutch phoneme set. The Afrikaans phonemes without matching Dutch seed model are given in Table 6.3 together with the respective manually chosen Dutch seed model. Furthermore, since the diphthongs iə, uə, əu, əi are not present in the Dutch phoneme set, we split them into individual phonemes (monophthongs) as it was also done by Engelbrecht and Schultz [2005].

6.4.3 Maximum a posteriori (MAP) adaptation

Since Köhler [1998] has shown that MAP adaptation is suitable for cross-lingual acoustic model adaptation, we also evaluate MAP adaptation. More specifically, the mean $\boldsymbol{\mu}_m^\ell$ of mixture component m and state ℓ is adapted as follows:

$$\hat{\boldsymbol{\mu}}_m^\ell = \frac{N_m^\ell}{N_m^\ell + \tau}\boldsymbol{\mu}_m^{A,\ell} + \frac{\tau}{N_m^\ell + \tau}\boldsymbol{\mu}_m^{D,\ell}, \tag{6.4}$$

where N_m^ℓ is the occupation likelihood of the Afrikaans data, τ a parameter to tune, $\boldsymbol{\mu}^A$ the mean of the Afrikaans data and $\boldsymbol{\mu}^D$ the mean of the Dutch data.

As seed models, we use the same Dutch context-dependent HMM/GMM models as in Section 6.4.2. For Afrikaans phonemes without a matching Dutch seed model, we again map phonemes as explained in Section 6.4.2 and Table 6.3.

6.4.4 Tandem

For the Tandem system, as done with the HMM/GMM system, we train context-independent monophone models that serve as seed models for the three-state context-dependent triphone models. We use a mixture of eight Gaussians per state to model the emission probabilities. Usually PCA reduces the dimensionality of the feature vectors to about 30 [Qian et al., 2011, Grézl et al., 2011]. Our initial study on Afrikaans revealed that the dimensionality of the feature vectors greatly affects the performance of the Tandem system [Imseng et al., 2012c]. Furthermore, we observed that preserving 99% of the variance yielded similar results to using all the dimensions. Therefore, we fix the dimensionality of the feature vectors such that 99% of the variance is preserved. Note that the feature dimensionality of different systems varies and is given in Tables 6.4, 6.5 and 6.6.

To balance the number of parameters with the amount of available training data, we also use the MDL-based decision tree [Shinoda and Watanabe, 1997].

6.4.5 KL-HMM

For the KL-HMM system, as also done for HMM/GMM and Tandem, we train context-independent monophone models that serve as seed models for the three-state context-dependent triphone models.

For KL-HMM, we apply the decision tree clustering reformulated as dictated by the KL-divergence criterion, presented in Section 4.6.2, page 55. Since it is not obvious how to apply the MDL principle to the modified clustering approach, we tune the threshold that determines the number of tied states on the cross-validation set.

6.4.6 Subspace Gaussian mixture models (SGMM)

The training of SGMMs is also done from context-independent monophone models that serve as seed models for the three-state context-dependent triphone models.

Decision tree clustering is done automatically, after having specified the number of leaves to be similar to the Tandem system. The UBM has $I = 500$ Gaussians and the dimensionality of the substate phone-specific vectors, U, is fixed to 50.

Table 6.4: Using 3 h of Afrikaans data to build a monolingual ASR system. The bi-gram phoneme model has a phoneme perplexity of 14.7 on the test set. Acoustic modeling techniques are described in Section 6.4. The best result is marked bold; italic numbers point to results that are not significantly worse.

System	Feature dimension	Number of tied states	Phoneme accuracy
HMM/GMM	39	1,447	61.2 %
KL-HMM	187	15,207	60.6 %
Tandem	48	1,846	*64.7* %
SGMM	39	2,000	**65.5** %

6.5 Evaluation

In this section, we analyze the performance of the different systems. We always apply the same bi-gram phoneme model as described in Section 6.2.1 and report Afrikaans phoneme accuracies on the test set (about 50 min of data). The bi-gram phoneme model scaling factor is determined for each system independently on the cross-validation set (see Section 6.3.1). In general, we expect that the exploitation of Dutch data will improve the Afrikaans ASR performance.

6.5.1 Afrikaans data only

For the first set of experiments, we only use the Afrikaans training set (3 h of data) for the training of the global and local parameters. More specifically, the MLP for the posterior feature extraction as well as the globally shared SGMM parameters are trained on three hours of Afrikaans (see Table 6.2 for MLP details). In previous studies [Povey et al., 2010], SGMM outperformed HMM/GMM when 15 h of training data was used. We hypothesize that SGMM also outperforms conventional HMM/GMM if only three hours of data is available for training. Furthermore, we hypothesize that Tandem outperforms conventional HMM/GMM and KL-HMM systems if three hours of Afrikaans data is available for training.

Results achieved by the different systems are summarized in Table 6.4. At the start of our work, the only baseline results available were from van Heerden et al. [2009], reporting 63.1% phoneme accuracy. However, the official train and test set were compiled after the official database release. Personal communication with the HLT group at Meraka confirmed that the lower performance of our baseline can be attributed to the different data partitioning. The HLT group now also uses the partitioning that we use for these experiments, and also report a lower performance.

The results in Table 6.4 confirm our hypotheses. On Afrikaans data only, SGMM performs best, followed by Tandem. Bold numbers in tables mark the best result and italic numbers point to results that are not significantly different from the best performance (see Section

2.4 for details about the significance test). KL-HMM and the HMM/GMM baseline perform significantly worse than SGMM and Tandem.

Table 6.4 also shows the feature dimensionality of the employed acoustic modeling techniques. HMM/GMM and SGMM are both based on MF-PLP features (39 dimensions). KL-HMM uses the raw output of the Afrikaans MLP. For the Tandem system however, recall that the posterior features need to be post-processed. Keeping 99% of the variance after PCA results in a 48-dimensional feature vector.

The number of tied states, also shown in Table 6.4, for HMM/GMM and for Tandem are automatically determined with the MDL criterion. We fix the number of tied states for the SGMM system similar to the number of tied states for the Tandem system. The number of tied states for the KL-HMM is tuned on the cross-validation set. Since the categorical distributions of the KL-HMM can be trained with very few data, modeling each triphone state separately performs best on the cross-validation set. Hence, the decision tree is only used to model unseen triphone contexts during testing.

Due to the extremely high number of states of the KL-HMM system, compared to the other systems, the KL-HMM system has the most parameters of the four systems given in Table 6.4. To explore whether an increased number of parameters improves the performance of the other systems, we increased the number of Gaussians per state for the HMM/GMM as well as for the Tandem system to 16 and doubled the number of sub-states of the SGMM system. However, this did not yield any improvement for any of the systems.

6.5.2 Auxiliary Dutch data

Since three hours seems to be a reasonable amount of training data, we also simulate very low-resourced languages and evaluate three different scenarios: six minutes of data, one hour of data and three hours of data. For comparison, we also evaluate a conventional HMM/GMM system for each scenario. We hypothesize, that KL-HMM performs best for very low amounts of data because we have seen this behavior in previous similar evaluations of KL-HMM [Imseng et al., 2012d]. If three hours of data is available, we expect that KL-HMM, Tandem and SGMM are successfully exploiting the out-of-language data and performing similarly well.

Table 6.5 confirms our hypotheses. The HMM/GMM (only trained on Afrikaans) is clearly outperformed by KL-HMM, Tandem and SGMM, hence all three systems successfully exploit out-of-language information. MLLR and MAP, however, only perform better than HMM/GMM if six minutes of Afrikaans data are available. Similar to the study on Greek data (see Figure 4.4, page 60), MAP outperforms MLLR if there is 1 h or more data available. Note that both approaches are bound to phoneme sets. Köhler [1998], for example, used a multilingual seed model that was trained from data associated with a matching IPA symbol for each phoneme. In our case however, we need to manually map several Afrikaans phoneme models as discussed in Table 6.3. Furthermore, MAP and MLLR may both suffer from the fact that the Dutch

Table 6.5: Exploiting Dutch data to improve an Afrikaans ASR system. The acoustic modeling techniques are described in Section 6.4. TS stands for the number of tied states, PA for phoneme accuracy in percent and τ is the parameter of the MAP adaptation. Best results of each PA column are marked bold; italic numbers point to results that are not significantly worse.

System	Feat.	6 min			1 h			3 h		
	dim.	TS	τ	PA	TS	τ	PA	TS	τ	PA
HMM/GMM	39	116	—	38.6	594	—	55.3	1,447	—	61.2
MLLR	39	—	—	41.3	—	—	44.4	—	—	44.7
MAP	39	11,357	15	39.4	11,357	5	46.9	11,357	1	50.6
KL-HMM	1,789	635	—	**53.1**	13,308	—	**61.5**	15,207	—	*67.3*
Tandem	286	114	—	41.0	537	—	*61.3*	1,846	—	*68.2*
SGMM	39	150	—	40.2	600	—	*60.4*	2,000	—	**68.5**

decision tree does not represent the context of Afrikaans very accurately. That problem could be further addressed with the polyphone decision tree specialization algorithm [Schultz and Waibel, 2000]. Wang et al. [2003] for example, successfully combined MAP adaptation and the polyphone decision tree algorithm on a non-native ASR task.

For the three hours, as well as the one hour scenario, SGMM, KL-HMM and Tandem all perform statistically the same. While SGMM is the most suitable acoustic modeling technique if we train only on within-language data, KL-HMM (which was performing significantly worse in Table 6.4) benefits most from out-of-language data, and seems to be particularly well suited to exploit out-of-language information on this database. Furthermore, KL-HMM using six minutes of data performs almost as well as a conventional monolingual HMM/GMM system using one hour of data. In the case of SGMMs, results are slightly worse than expected. We suppose that the dimensionality of the sub-state-specific vectors is probably too high for only six minutes of data.

6.5.3 Within- and out-of-language data

We have already shown that properly combining acoustic information from multiple similar languages can boost the performance. Therefore, we hypothesize that the performance can be improved if we concatenate the output of both MLPs or train the shared SGMM parameters on both languages. The concatenated MLP outputs are renormalized to guarantee that the feature vectors can be interpreted as posterior distributions, as assumed by the KL-HMM. For the Tandem systems, we post-process the normalized vectors as already described in Section 6.4.4. For SGMM, we train the shared parameters with the data of both languages.

However, Table 6.6 shows that the results only marginally improve for Tandem and SGMM. For KL-HMM, they improve by 1.5% absolute. KL-HMM yields the best performance, but it is not statistically different from the performance of the other systems.

Table 6.6: Using the Dutch and Afrikaans MLP (KL-HMM and Tandem) and use Dutch and Afrikaans data to train the shared parameters (SGMM). The best result is marked bold; italic numbers point to results that are not significantly worse.

System	Feature dimension	Phoneme accuracy
KL-HMM	1,976	**68.8** %
Tandem	308	*68.4* %
SGMM	39	*68.6* %

6.6 Discussion

The results in Section 6.5 have shown that (a) out-of-language data improved an existing Afrikaans speech recognizer and (b) KL-HMM outperforms all other approaches if only 6 min of Afrikaans data are available. In this section, we discuss the two conclusions.

6.6.1 Improvement through out-of-language data

All systems in Table 6.6 perform significantly better than the HMM/GMM baseline that does not use Dutch data (see Table 6.4). We hypothesize that Dutch data mostly improves recognition accuracy of phonemes for which the Afrikaans dataset does not contain much training data. Figure 6.4 shows the relative phoneme accuracy change (relative gain) per phoneme of the systems given in Table 6.6 with respect to the HMM/GMM baseline that does not use Dutch data. The phonemes on the x-axis are sorted according to their frequency in the Afrikaans training data with the most frequent phonemes on the left. Figure 6.4 appears to confirm our hypothesis, since rare phonemes like *2* (ø in IPA) benefit more from the out-of-language data than frequent phonemes such as @ (ə in IPA).

6.6.2 Advantage of KL-HMM

Even though we performed an extensive error analysis, there is no clear evidence for which phonemes KL-HMM yields most improvement compared to the other modeling techniques. Rather, KL-HMM consistently improves the recognition accuracy across all phonemes. We attribute the improvement to the sophisticated acoustic modeling and the constrained optimization space that are particularly well suited for small amount of data scenarios.

6.7 Conclusion

We successfully exploited Dutch data and boosted a monolingual speech recognizer that was trained on three hours of Afrikaans data. We compared KL-HMM, Tandem, SGMM, MLLR as well as MAP adaptation and found that KL-HMM, Tandem and SGMM successfully exploit out-of-language data if at least one hour of within-language data are available. If only six

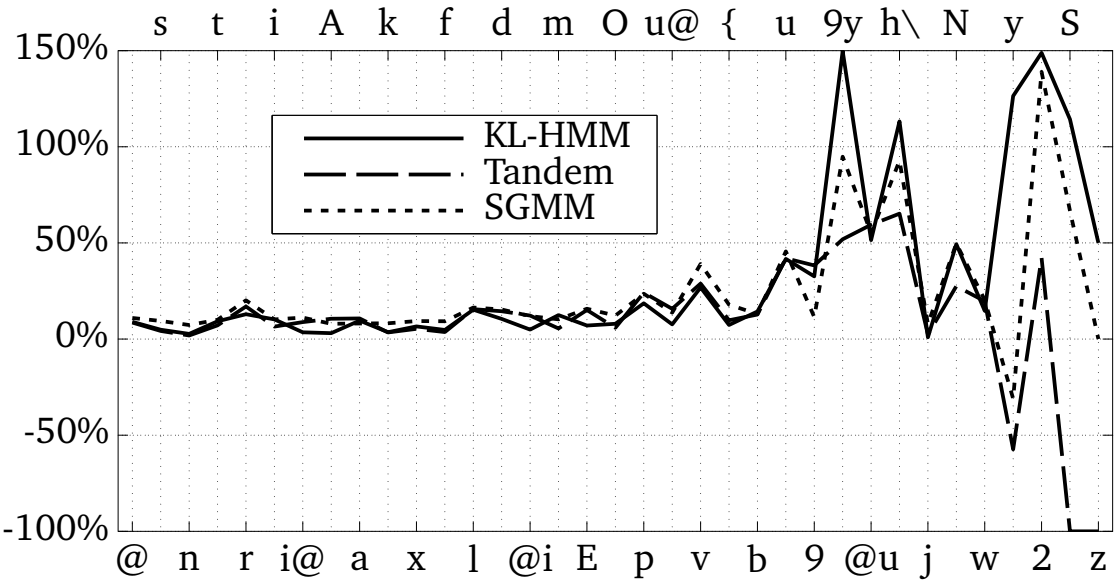


Figure 6.4: Relative phoneme accuracy change (relative gain) per phoneme of the systems shown in Table 6.6 with respect to the monolingual HMM/GMM baseline system. The phonemes on the x-axis are sorted according to their frequency in the Afrikaans training data (most frequent phoneme on the left). For better readability, the x-axis is labeled on the top and at the bottom of the figure.

minutes of data are available, KL-HMM outperforms all other acoustic modeling techniques including MLLR and MAP adaptation.

More specifically, we found that if three hours of within-language data and 80 hours of out-of-language data are available, the proposed systems yield 12% relative improvement compared to a conventional HMM/GMM system only using within-language data. If only six minutes of within-language data and 80 hours of out-of-language data are available, KL-HMM performs relatively about 30% better than MLLR and MAP adaptation.

7 Speaker adaptive KL-HMM

In the previous chapters, KL-HMM was successfully applied to accented and under-resourced speech recognition tasks in multilingual setups through efficient feature level adaptation and parsimonious use of parameters. This previous work suggests that we may also get improvement in monolingual scenarios using conventional techniques such as speaker adaptation, especially in the case of non-native speech. Therefore, in this chapter, inspired from MAP adaptation, we further boost KL-HMM performance by applying Bayesian speaker adaptation, directly applied to posterior features.

The speaker adaptive KL-HMM exploits the parsimonious use of parameters of KL-HMM that efficiently uses very limited amounts of training data. More specifically, speaker adaptive KL-HMM performs a simple, adaptive regression between phone posteriors estimated with an MLP on large amounts of speaker-independent training data, and speaker-specific phone posteriors generated by the speaker-independent MLP on very limited amount of speaker-specific adaptation data. Using Swiss French data from the MediaParl database (see Section 2.5, page 25), we show that such speaker adaptive KL-HMM significantly outperform conventional adaptation approaches such as MLLR and MAP on non-native speech.

7.1 Motivation

Several speaker adaptation techniques such as MLLR [Gales, 1998] or MAP adaptation [Gauvain and Lee, 1993] have been proposed to improve ASR performance. As we have already seen in Chapter 5, speaker adaptation is also particularly relevant in the case of non-native ASR, given the high variability of accented speech and the usually small amount of non-native speech data available for training [Wang et al., 2003, Bouselmi et al., 2008, Segura et al., 2007, Gemello et al., 2007]. In the context of HMM/GMM, conventional solutions include MLLR and MAP [Wang et al., 2003, Segura et al., 2007]. In the case of hybrid HMM/MLP systems, an LHN was typically used to adapt the MLP to a speaker [Gemello et al., 2007].

We have seen in Chapter 5, that KL-HMM in a multilingual setup can outperform MLLR and

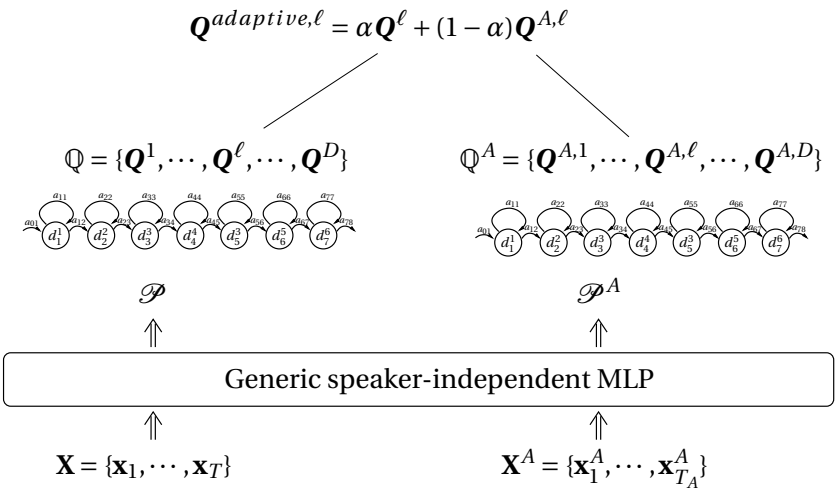


Figure 7.1: Illustration of speaker adaptive KL-HMM. The generic data $\boldsymbol{X}$ and the speaker-specific adaptation data $\boldsymbol{X}^A$ are both passed through the speaker-independent MLP and used to train a speaker-independent and a speaker-dependent KL-HMM. The categorical distributions are then combined at the state-level.

LHN for non-native speaker adaptation. In those experiments, it was also observed that KL-HMM was quickly yielding state-of-the-art performance with limited amount of training data. More specifically, on the HIWIRE database, that contains spoken pilot orders, we also trained and evaluated speaker dependent KL-HMM systems, i.e. a KL-HMM trained on the data of a single speaker only (see Table 5.6, page 72).

In this chapter, we go one step further and investigate speaker adaptive KL-HMM on data from the bilingual MediaParl database [Imseng et al., 2012a]. MediaParl is a Swiss accented bilingual database containing recordings in both accented French and German, as they are spoken at the Parliament in Valais, a state of Switzerland (see also Section 2.5, page 25). The advantage of MediaParl is that it is a pretty large multilingual database and the test set consists of bilingual speakers, hence non-native and native speech recorded at same conditions.

Similar to MAP adaptation in HMM/GMM based ASR systems, that adapts the means of the GMMs, the proposed speaker adaptive KL-HMM adapts the generic reference posteriors of the KL-HMM. Just using a couple of minutes of speech data, and using the same speaker-independent MLP to generate features, we train a speaker-specific KL-HMM. The generic KL-HMM reference posteriors are then adapted by performing a linear combination with the speaker-specific reference posteriors.

In the following section, we will first introduce the speaker adaptive KL-HMM concept. Then, experimental setup and results are given in Sections 7.3 and 7.4, respectively.

7.2 Speaker adaptive KL-HMM

The results in Table 5.6, page 72 suggest that KL-HMM performs extremely well when only a small amount of training data is available. Even though it is not an adaptation technique, the categorical distributions are trained and not adapted, it can outperform current state-of-the-art adaptation techniques such as MLLR and LHN based speaker adaptation. However, if the amount of data to train/adapt gets below a certain threshold, KL-HMM may overfit. Therefore, we introduce the concept of speaker adaptive KL-HMM, also illustrated in Figure 7.1.

We assume to have a generic KL-HMM system with a set of already trained categorical distributions, $\mathbb{Q} = \{\boldsymbol{Q}^1, \cdots, \boldsymbol{Q}^\ell, \cdots, \boldsymbol{Q}^D\}$, where each categorical $\boldsymbol{Q}^\ell$ is associated with a tied state and D is the total number of tied states. The categorical distributions $\mathbb{Q}$ have been trained from speaker-independent generic data $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$.

Furthermore, we suppose to have a small amount of transcribed speaker-specific adaptation data $\boldsymbol{X}^A = \{\boldsymbol{x}_1^A, \cdots, \boldsymbol{x}_{T_A}^A\}$, where $T_A \ll T$. Given the speaker-specific data $\boldsymbol{X}^A$, we can generate the posterior features sequence $\mathscr{P}^A = \{\boldsymbol{P}_1^A, \cdots, \boldsymbol{P}_{T_A}^A\}$ by using the same speaker-independent MLP as used to generate $\mathscr{P}$. The posterior features sequence $\mathscr{P}^A$ together with transcriptions can then be used to train a speaker specific KL-HMM with categorical distributions $\mathbb{Q}^A = \{\boldsymbol{Q}^{A,1}, \cdots, \boldsymbol{Q}^{A,\ell}, \cdots, \boldsymbol{Q}^{A,D}\}$ along the same procedure as $\mathbb{Q}$ was obtained. For the speaker-specific KL-HMM training, we use the generic categorical distributions $\mathbb{Q}$ as seed models (i.e. initialization: $\mathbb{Q}^A = \mathbb{Q}$).

However, due to the small amount of adaptation data, we expect the speaker-specific KL-HMM parameters $\mathbb{Q}^A$ to overfit. To overcome that problem, we combine the generic $\mathbb{Q}$ and the speaker specific $\mathbb{Q}^A$ at the state-level:

$$\boldsymbol{Q}^{\text{adaptive},\ell} = \alpha \boldsymbol{Q}^\ell + (1-\alpha)\boldsymbol{Q}^{A,\ell}, \tag{7.1}$$

where $\boldsymbol{Q}^{\text{adaptive},\ell}$ stands for the categorical distribution of the speaker adaptive KL-HMM and $\alpha \in [0,1]$ is a parameter of the combination.

7.3 Experimental setup

We evaluate the speaker adaptive KL-HMM on French MediaParl data and compare it to standard KL-HMM, a conventional HMM/GMM system and MAP and MLLR adaptation.

7.3.1 Data

For our studies, we use the French part of the MediaParl database [Imseng et al., 2012a]. MediaParl is a Swiss accented bilingual database containing recordings in both French and German as they are spoken in Switzerland. The data were recorded at the Valais Parliament.

The MediaParl database contains a dictionary with all the words (no out of vocabulary words) and standardized training, development and test sets as described in Section 2.5, page 25. The bigram language model that we use for this study (see Table 7.1) is trained on two sources: the transcriptions of the training set and texts from the corpus Europarl, a multilingual corpus of European Parliament proceedings [Koehn, 2005]. Europarl is made up of about 50 million words for each language and is used to overcome data sparsity of the MediaParl texts. However, the vocabulary is limited to the sole words from MediaParl.

The test set, shown in Table 7.2, contains all the seven speakers that speak in both languages. In this chapter, we study fast speaker adaptation (minutes of data for each speaker) on the French part of the data. *Speaker 059* is discarded because a couple of French phonemes are not pronounced at all. For all the other speakers, we randomly select five minutes of adaptation data (and exclude that data from the test set). Only for *speaker 079* (2 minutes of French data in total) we use about half the data for adaptation and the other half for testing.

7.3.2 Systems

We investigate five systems: conventional HMM/GMM, MLLR, MAP adaptation, KL-HMM and speaker adaptive KL-HMM.

HMM/GMM

The standard HMM/GMM system does not use the adaptation data. We use the training data from the French MediaParl corpus to train a conventional crossword context-dependent speech recognizer. Each triphone is modeled with three states from which each one is modeled with 16 Gaussians. To tie rare states, we apply a decision tree clustering. The MDL criterion is used to determine the number of tied states [Shinoda and Watanabe, 1997]. For decoding, we use the bigram language model as described in Section 7.3.1 and tune the language model scaling factor as well as the word insertion penalty on the development data.

MLLR

In one of our studies, we investigated MLLR as well as a constrained version of it (CMLLR) to evaluate whether a new language could be accommodated by linear transforms [Imseng et al., 2012b]. CMLLR has fewer parameters and we assumed that this could be advantageous if we only have access to a limited amount of data. However, even if we only used 5 minutes of

Table 7.1: Properties of the French language model: number of words, number of bigrams and perplexity on the development and test set.

Vocabulary size	Number of bigrams	Perplexity on DEV	Perplexity on TST
12,035	1.5 M	147	152

Table 7.2: MediaParl-TST: speakers using both languages form the test set. For each speaker the number of French and German sentences is given.

Speaker	Sentences in French	Adapt data [min]	Test [min]	Sentences in German	Mother tongue
059	31	-	-	195	German
079	22	1	1	698	German
094	313	5	60	72	French
096	89	5	15	8	French
102	72	5	7	7	French
109	233	5	46	402	German
191	165	5	28	310	German
Total	925	26	157	1,692	

adaptation data, MLLR outperformed CMLLR. Therefore, in this study, we only investigate standard MLLR . For this, we use the adaptation data described in Table 7.2 to perform speaker adaptation and employ a regression tree that allows up to 16 regression classes.

MAP

Since speaker adaptive KL-HMM is very similar in spirit to MAP adaptation, we also investigate MAP based speaker adaptation on the French MediaParl data. Recall that the mean $\boldsymbol{\mu}_m^\ell$ of mixture component m and state ℓ is adapted as follows:

$$\hat{\boldsymbol{\mu}}_m^{adapted,\ell} = \frac{N_m^\ell}{N_m^\ell + \tau}\boldsymbol{\mu}_m^{A,\ell} + \frac{\tau}{N_m^\ell + \tau}\boldsymbol{\mu}_m^\ell, \tag{7.2}$$

where N_m^ℓ is the occupation likelihood of the speaker-specific adaptation data, τ a parameter to tune, $\boldsymbol{\mu}^A$ the mean estimated on the adaptation data and $\boldsymbol{\mu}$ the generic speaker-independent mean. The tuning of τ is discussed in the next section.

KL-HMM

For the standard KL-HMM system, we first train the generic speaker-independent MLP from the same 39 MF-PLP features that we used for the HMM/GMM system training.

Similar to the experiments presented in Chapter 6, the MLP is trained on triphone targets. To obtain triphone targets, we use the standard HMM/GMM system with a different decision tree. As described by [Shinoda and Watanabe, 1997], the MDL criterion has a hyper-parameter, c, which controls the weight of the term that penalizes models with large amounts of tied states. For the triphone target creation, we use $c = 16$, as also done in Chapter 6, page 79, to obtain 659 tied states, used as MLP targets.

We use all the French MediaParl training data to train a crossword context-dependent (tied

states) KL-HMM based speech recognizer. Similar to the HMM/GMM system, the standard KL-HMM system does not use the adaptation data. During state tying, we fix the minimum occupancy threshold to 20 and tune the minimum decrease in the cost function threshold on the development data. For decoding, we use the same bigram language model as for the HMM/GMM system and tune language model scaling factor and word insertion penalty on the development data.

Speaker adaptive KL-HMM

The speaker adaptive KL-HMM is trained as described in Section 7.2. As seed models, we use the KL-HMM system presented above. The tuning of the parameter α is discussed in the next section.

7.4 Results

In this section, we first discuss the tuning of the parameters α and τ for the speaker adaptive KL-HMM and MAP adaptation, respectively. Then we show that MLLR outperforms MAP adaptation on the investigated task and therefore subsequently compare the HMM/GMM, MLLR, KL-HMM and speaker adaptive KL-HMM against each other.

7.4.1 Tuning of the parameter α

As we discussed in (7.1), page 91, speaker adaptive KL-HMM involves the parameter α. Figure 7.2 shows the influence of α. If α is set to one, the speaker adaptive KL-HMM is equivalent to the standard KL-HMM. For each speaker, there is at least one α value for which the performance of the speaker adaptive KL-HMM is better than the performance of the standard KL-HMM. However, we also see that, for some values of α, the performance decreases. It can clearly be seen that α-values close to zero perform bad in general, i.e. the adapted KL-HMM system overfits. The highest performance gains can be seen for two non-native speakers (*079* and *191*). The French HMM/GMM baseline system reported in [Imseng et al., 2012a] performed particularly bad on these two speakers, hence they seem to have a strong accent. This hypothesis was verified by native speakers who listened to the utterances of these speakers.

During the system comparison, we will use the best performing α value that we found for each speaker (on the test set). The parameter tuning on the test set is suboptimal, but the low amount of data per speaker does not allow a separate development set. For the MAP based adaptation, we will also tune τ on the test set.

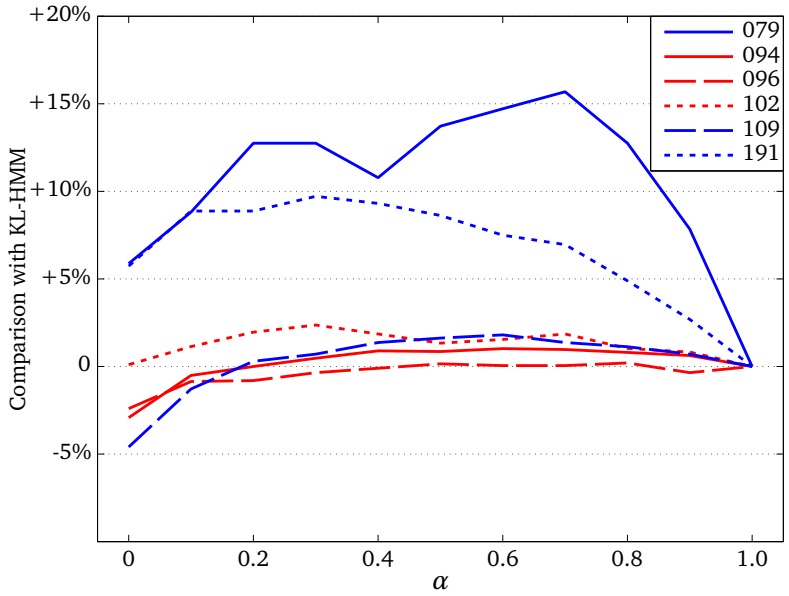


Figure 7.2: Relative improvement of speaker adaptive KL-HMM with respect to speaker independent KL-HMM (y-axis shows relative performance change). Each curve represents one speaker. Red curves represent native speakers and blue curves stand for non-native speakers. This figure also shows the impact of parameter α.

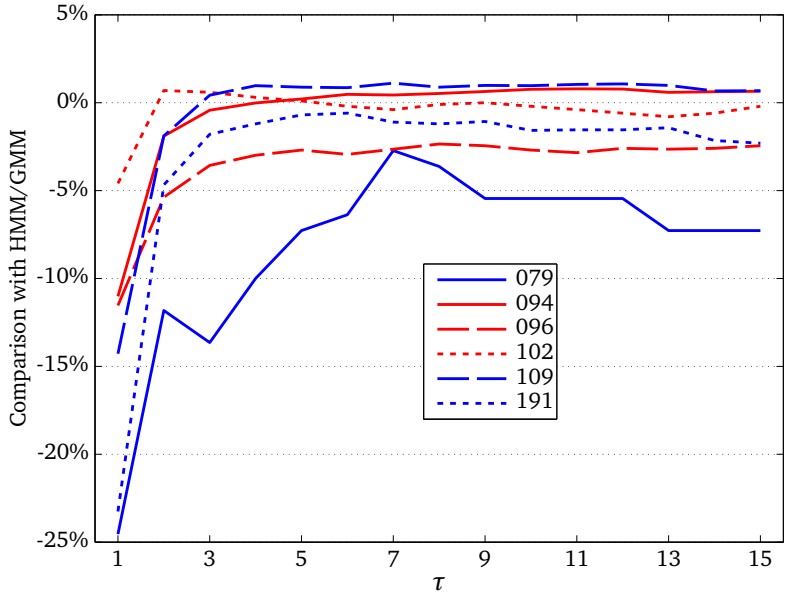


Figure 7.3: Relative performance change of MAP adaptation with respect to the standard HMM/GMM performance. Each curve represents one speaker. Red curves represent native speakers and blue curves stand for non-native speakers. This figure also shows the impact of parameter τ.

Table 7.3: Comparison of HMM/GMM, MAP adaptation and MLLR on French MediaParl data. The optimal parameter τ is used for MAP adaptation and MLLR results are based on a regression tree that allows up to 16 classes.

Speaker	079	094	096	102	109	191	Total
HMM/GMM	46.6%	79.7%	81.3%	79.1%	70.0%	59.0%	73.1%
MAP	45.3%	80.3%	79.4%	79.7%	70.8%	58.6%	73.3%
MLLR	47.5%	80.1%	80.4%	79.8%	70.5%	60.6%	73.6%

7.4.2 Tuning of the parameter τ

Recall from (7.2), page 93, that MAP adaptation makes use of the parameter τ. Figure 7.3 shows the influence of τ. During the under-resourced ASR study in the last chapter, we observed that the optimal τ for MAP adaptation on the Afrikaans task varied between 1 and 15 for 3 h and 6 min of data, respectively (see Table 6.5, page 86). Therefore, for this study, we tune τ in the interval $[1, 15]$ for each speaker separately (on the test) set and compare the performance of MAP adaptation with standard HMM/GMM systems. Since we only have 1 min of adaptation data for speaker *079*, that curve is the lowest and reaches the maximum at about 3% degradation compared to the unadapted HMM/GMM system.

In Table 7.3, we compare the HMM/GMM system to MAP adaptation with manually tuned τ, i.e. optimal value in $[1, 15]$ is determined for each speaker, and MLLR. Overall, MAP only improves marginally over unadapted HMM/GMM and MLLR performs best. These findings are consistent with the study on Afrikaans in the previous chapter and an earlier study on non-native ASR [Wang et al., 2003], where MLLR also performed better than MAP for low amounts of data (less than 10 min). For the system comparison in the next section, we therefore only use the MLLR results as a representative of conventional adaptation approaches for HMM/GMM based systems.

7.4.3 System comparison

In Figure 7.4, the performance of a standard HMM/GMM system, MLLR, KL-HMM and speaker adaptive KL-HMM are compared. In the plot on the left and in the center plot, the performance on native and non-native speech, respectively, is shown. White bars represent HMM based systems (No Adapt=HMM/GMM, Adapt=MLLR) and colored bars represent KL-HMM based systems (No Adapt=KL-HMM, Adapt=speaker adaptive KL-HMM). At first glance, we observe that for native speech, the HMM/GMM based systems perform better and for non-native speech, the KL-HMM based systems perform better. If we have a closer look, we can distinguish four different cases:

- *No adapt* on native speech: the HMM/GMM system performs significantly better than the KL-HMM system

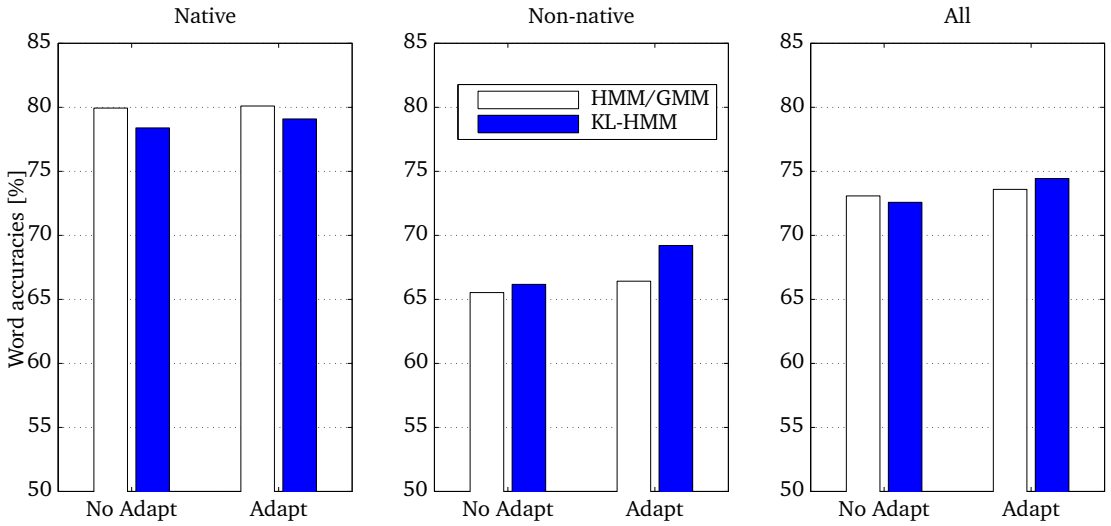


Figure 7.4: Comparison of HMM/GMM, MLLR, KL-HMM and speaker adaptive KL-HMM on French MediaParl data. The left and center plot shows word accuracies on native and non-native speech, respectively. The right plot shows word accuracies on all the test data (native and non-native speech). *No Adapt* stands for HMM/GMM and KL-HMM (not used the adaptation data) and *adapt* stands for MLLR and speaker adaptive KL-HMM (used the adaptation data).

- *Adapt* on native speech: MLLR performs significantly better than the speaker adaptive KL-HMM system, but the gap between HMM/GMM and KL-HMM is smaller than for the standard systems
- *No adapt* on non-native speech: there is no significant difference between the HMM/GMM and the KL-HMM system
- *Adapt* on non-native speech: the speaker adaptive KL-HMM performs significantly better than MLLR

As seen in the right plot of Figure 7.4, the speaker adaptive KL-HMM system yields the best overall performance.

7.5 Conclusion

In this chapter, we introduced a speaker adaptation approach for KL-HMM. Fast speaker adaptation is achieved by exploiting the parsimonious use of KL-HMM parameters, which efficiently use very limited amounts of training data. Reference KL-HMM categorical distributions are then expressed as a linear combination of phone posteriors estimated on large amounts of speaker-independent training data, and speaker-specific phone posteriors obtained on very limited amount of speaker-specific adaptation data. On non-native Swiss French data, the speaker adaptive KL-HMM has been shown to significantly outperform MLLR.

8 Conclusion and future directions

8.1 Conclusion

In this thesis, we tackled acoustic modeling issues related to multilingual adaptation and lexical diversity across databases. In the context of hybrid HMM/MLP, we elaborated on KL-HMM acoustic modeling, a parsimonious modeling technique that parametrizes the HMM states with reference posteriors estimated along a principled algorithm using a Kullback–Leibler divergence based cost function that is suitable for posterior distributions. In that context, we showed how to train the HMM and the MLP parameters on different databases. We extended the recently proposed KL-HMM approach by a decision tree clustering algorithm allowing us to build a recognizer based on triphones and integrated high dimensional posterior features, estimated by an MLP trained on context-dependent targets. In this vein, the MLP can be trained on large amounts of data in any language and optimally utilize the data by adjusting the number of MLP outputs. More MLP outputs allow a more subtle distinction of acoustic samples. The HMM, on the other hand, can be trained on low amounts of target language data thanks to the decision tree clustering that allows parameter sharing through state tying.

In the context of accented and under-resourced speech recognition, involving high acoustic phone variability, mismatches between phone sets of multiple languages and small amounts of data, the resulting speech recognition system has been shown to be able to efficiently exploit multilingual training data. In the case of accented speech recognition, this conclusion is further supported by additional evidence and theoretical and experimental comparisons with similar approaches such as probabilistic acoustic mapping, linear hidden networks and MLLR. Furthermore, for under-resourced ASR tasks, we successfully exploited Dutch data and boosted a monolingual Afrikaans speech recognizer. We also compared KL-HMM with Tandem and SGMM and found that all three acoustic modeling techniques successfully exploit out-of-language data if at least one hour of within-language data are available. However, if less training data is available, KL-HMM outperforms the other acoustic modeling techniques, including cross-language MLLR and MAP based adaptation.

Finally, the KL-HMM framework was further extended by a speaker adaptation method,

referred to as speaker adaptive KL-HMM. Speaker adaptive KL-HMM allows the expression of the reference posteriors as a linear regression between reference vectors trained on posterior features estimated on large amounts of speaker-independent training data, and reference vectors trained on speaker-specific posterior features obtained from very limited amount of speaker-specific adaptation data. Validation experiments on non-native Swiss French data showed that the speaker adaptive KL-HMM is able to significantly outperform conventional MLLR and MAP based speaker adaptation.

In conclusion, the KL-HMM framework has been shown to be a suitable alternative to conventional acoustic modeling techniques and seems to be preferable in low amount of data as well as phoneme set mismatch scenarios. However, for well resourced languages, KL-HMM seems to be outperformed by current acoustic modeling techniques such as SGMMs. Potential drawbacks of the KL-HMM framework may be the rather small number of parameters and the absence of an upper bound for the Kullback–Leibler divergence. The latter may, during decoding, theoretically result in different dynamic ranges for the local scores of different HMM state distributions.

8.2 Potential future research directions

Multilingual environments, such as the bilingual parliament of the Swiss state Valais, involve numerous challenges for state-of-the-art ASR systems. We mainly addressed multilingual acoustic modeling issues and showed how to exploit multilingual acoustic training data to improve the performance of ASR systems in the case of non-native speech and speech from under-resourced languages. How to efficiently handle code-switches remains a very challenging research problem. Hence, to fully integrate ASR systems in multilingual environments, multilingual language modeling issues should be further investigated. Furthermore, multilingual language models also allow the exploitation of multilingual pronunciation dictionaries. The presented KL-HMM framework also seems to have potential for the creation of a dictionary from scratch, or for the integration of pronunciation variants [Rasipuram and Magimai.-Doss, 2012]. It may therefore be useful to tackle multilingual pronunciation dictionary issues.

The simple acoustic model structure and the possibility to directly model posterior features are valuable properties that suggest the further development of the KL-HMM framework. Potential future research directions include the study of longer contexts, enabling to increase the number of parameters, the extension of the current Viterbi training algorithm into a full expectation-maximization algorithm and the reconsideration of the Kullback-Leibler divergence, which potentially may be replaced by similar measures that have an upper bound such as the Jensen–Shannon divergence. Furthermore, the combination of discriminative and generative techniques, as done for example in Tandem systems, seems to be beneficial. Although the KL-HMM framework is principled, it is not a generative model. Initial investigations along that research direction [Garner and Imseng, 2013] led to different acoustic modeling techniques and revealed issues that need to be further addressed.

A Phoneme sets and manual mappings

Table A.1 shows all the phoneme sets used in this thesis. Each database that we used, comes with its dictionary that is build upon a particular phoneme set. To simplify the comparison, we converted SAMPA and Arpabet symbols into IPA format.

We present the phonemes in three categories, consonants (cons), vowels (vow) and diphthongs (diph). The consonants are sorted according to the following list: nasals, plosives, fricatives, approximants, lateral approximants, coarticulated consonants and then affricates. Within the same consonant category, phonemes are listed according to the place of articulation: labial, coronal, dorsal, radical and glottal. For the listing of the vowels, we follow the vowel quadrilateral from front to back (left to right) and from close to open (top to bottom).

There are two peculiarities:

- SAMPA Italian: the obstruents in Italian are classified along two dimensions, voiced versus voiceless and single versus geminate. The SAMPA based dictionary that comes with the Italian SpeechDat(II) database considers single and geminate consonants as different phonetic symbols. The geminate variants (GV) are listed separately in Table A.1.
- SAMPA French: when they are not functional there is a strong tendency in unstressed syllables towards indetermination. Indeterminacy symbols (IS) are listed separately in Table A.1.

Table A.2 then compares knowledge driven manual mapping and data-driven (hard decision mapping) of the target phonemes employed by the HIWIRE database to the English (EN) and universal (UNI) source phonemes used by the SpeechDat(II) database. Gray cells point to symbols that are different from the target phoneme symbol.

Figure A.1 shows the full international phonetic alphabet (IPA) as an additional reference.

Table A.1: Phoneme sets of all the databases used in this thesis. For the ease of comparison, all the phonemes are shown in IPA format. Each database also uses *sil* for silence (not shown in the table). The last column shows the total number of phonemes (*S*).

Phoneme set		Phonemes	*S*
Lwazi	Cons	m, n, ŋ, p, b, t, d, k, g, f, v, s, z, ʃ, ʒ, x, ɦ, r, j, l, w	
Afrikaans	Vow	i, y, u, øː, ə, ɛ, œ, ɔ, æ, a, ɑː	38
	Diph	iə, uə, əu, əi, œy	
CGN	Cons	m, n, ɲ, ŋ, p, b, t, d, k, g, f, v, s, z, ʃ, ʒ, x, ɣ, h, r, j, l, w	
Dutch	Vow	i, y, u, ɪ, ʏ, ỹ, ʏː, eː, øː, oː, ə, ɛ, ɛ̃, ɛː, ɔ, ɔ̃, ɔː, aː, ɑ, ɑ̃	47
	Diph	ɛi, œy, ɑu	
HIWIRE	Cons	m, n, ŋ, p, b, t, d, k, g, f, v, θ, ð, s, z, ʃ , h, r, j, l, w, tʃ, dʒ	
non-native	Vow	i, u, ɪ, ɛ, ɜ˞, ʌ, ɔ, æ, ɑ	38
English	Diph	eɪ, oʊ, ɔɪ, aʊ, aɪ	
MediaParl	Cons	m, n, ɲ, p, b, t, d, k, g, f, v, s, z, ʃ, ʒ, ʁ, ɥ, j, l, w	37
French	Vow	i, y, u, e, ẽ, ø, o, õ, ə, ɛ, œ, ɔ, a, ã, œ̃, ɐ	
MediaParl	Cons	m, n, ŋ, p, b, t, d, k, g, ʔ, f, v, s, z, ʃ, ʒ, ç, x, ʁ, h, r, j, l	
German	Vow	i, iː, y, yː, u, uː, ɪ, ʏ, ʊ, e, eː, ẽ, ø, øː, o, oː, õ, õː, ə, ɛ, ɛ̃, ɛ̃ː,ɛː, œ, ɔ, a, aː, ã, ãː, ɐ	57
	Diph	ɔʏ, aʊ, aɪ	
SpeechDat(II)	Cons	m, n, p, b, t, d, c, ɟ, k, g, f, v, θ, ð, s, z, ç, x, ɣ, r, j, ʝ, l, ts, dz	31
Greek	Vow	i, u, e, o, a	
SpeechDat(II)	Cons	m, n, ŋ, p, b, t, d, k, g, f, v, θ, ð, s, z, ʃ, ʒ, h, r, j, l, w, tʃ, dʒ	
English	Vow	iː, uː, ɪ, ʊ, e, ə, ɜː, ʌ, ɔː, æ, ɑː., ɒ	45
	Diph	ɪə, ʊə, eə, eɪ, əʊ, ɔɪ, aʊ, aɪ	
SpeechDat(II)	Cons	m, n, ɲ, ŋ, p, b, t, d, k, g, β, f, θ, ð, s, z, x, ɣ, r, j, ʝ, ɾ, l, ʎ, w, tʃ	32
Spanish	Vow	i, u, e, o, a	
SpeechDat(II)	Cons	m, n, ɲ, p, b, t, d, k, g, f, v, s, z, ʃ, r, j, l, ʎ, w, ts, dz, tʃ,dʒ	
Italian	GV	mm, nn, ɲɲ, pp, bb, tt, dd, kk, gg, ff, vv, ss, ʃʃ, rr, ll, ʎʎ, tts, ddz, ttʃ, ddʒ	52
	Vow	i, u, e, o, ə, ɛ, ɔ, a	
SpeechDat(II)	Cons	m, n, ɲ, ŋ, p, b, t, d, k, g, f, v, s, z, ʃ, ʒ, ʁ, ɥ, r, j, l, w	
Swiss French	Vow	i, y, u, e, ẽ, ø, o, õ, ə, ɛ, œ, ɔ, a, ã, œ̃, ɑ	42
	IS	œ/= ø or œ, ɛ/ = e or ɛ, ɔ/= o or ɔ	
SpeechDat(II)	Cons	m, n, ŋ, p, b, t, d, k, g, ʔ, f, v, s, z, ʃ, ʒ, ç, x, ʁ, h, j, l, ts, tʃ, pf	
Swiss German	Vow	iː, yː, uː, ɪ, ʏ, ʊ, eː, øː, oː, ə, ɛ, œ, ɔ, a, aː	59
	Diph	iːɐ, yːɐ, uːɐ, ɪɐ, ʊɐ, eːɐ, øːɐ, oːɐ, ɛɐ, ɛːɐ, œɐ, ɔɐ, ɔʏ, aʊ, aɐ, aːɐ, aɪ	

Table A.2: Knowledge driven, manual mapping and data-driven (hard decision mapping) of the target phonemes (HIWIRE) to the English (EN) and universal (UNI) source phonemes (SpeechDat(II)). Gray cells point to symbols that are different from the target phoneme symbol.

HIWIRE	UNI			EN	
	man	hard (UNI)	hard (sUNI)	man	hard (EN)
m	m	m	m	m	m
n	n	nn	n	n	n
ŋ	ŋ	ŋ	ŋ	ŋ	ŋ
p	p	pp	p	p	p
b	b	bb	b	b	b
t	t	tt	t	t	t
d	d	dd	d	d	d
k	k	k	k	k	k
g	g	g	g	g	g
f	f	f	f	f	f
v	v	v	v	v	v
θ	θ	pf	pf	θ	θ
ð	ð	ð	ð	ð	ð
s	s	ss	ss	s	s
z	z	dz	ʒ	z	z
ʃ	ʃ	ʃʃ	ʃʃ	ʃ	ʃ
h	h	h	h	h	h
r	r	r	r	r	r
j	j	jj	ʎ	j	j
l	l	ll	ll	l	l
w	w	w	w	w	w
tʃ	tʃ	tʃ	tʃ	tʃ	tʃ
dʒ	dʒ	dʒ	dʒ	dʒ	dʒ
i	i	iː	iː	iː	iː
u	u	uː	uː	uː	uː
ɪ	ɪ	ɪ	ɪ	ɪ	iː
ɛ	ɛ	eə	eə	e	eə
ɜˠ	ɜː	œ	œ	ɜː	ɜː
ʌ	ʌ	ẽ	aːɐ	ʌ	ɑː
ɔ	ɔ	oːɐ	ɔɐ	ɔː	ɔː
æ	æ	æ	æ	æ	æ
ɑ	ɑ	aː	aː	ɑː	ɑː
eɪ	eɪ	eː	eː	eɪ	eɪ
oʊ	əʊ	oː	oː	əʊ	ɔː
ɔɪ	ɔɪ	ɔɪ	ɔɪ	ɔɪ	ɔɪ
aʊ	aʊ	aʊ	aʊ	aʊ	aʊ
aɪ	aɪ	aɪ	aɪ	aɪ	aɪ

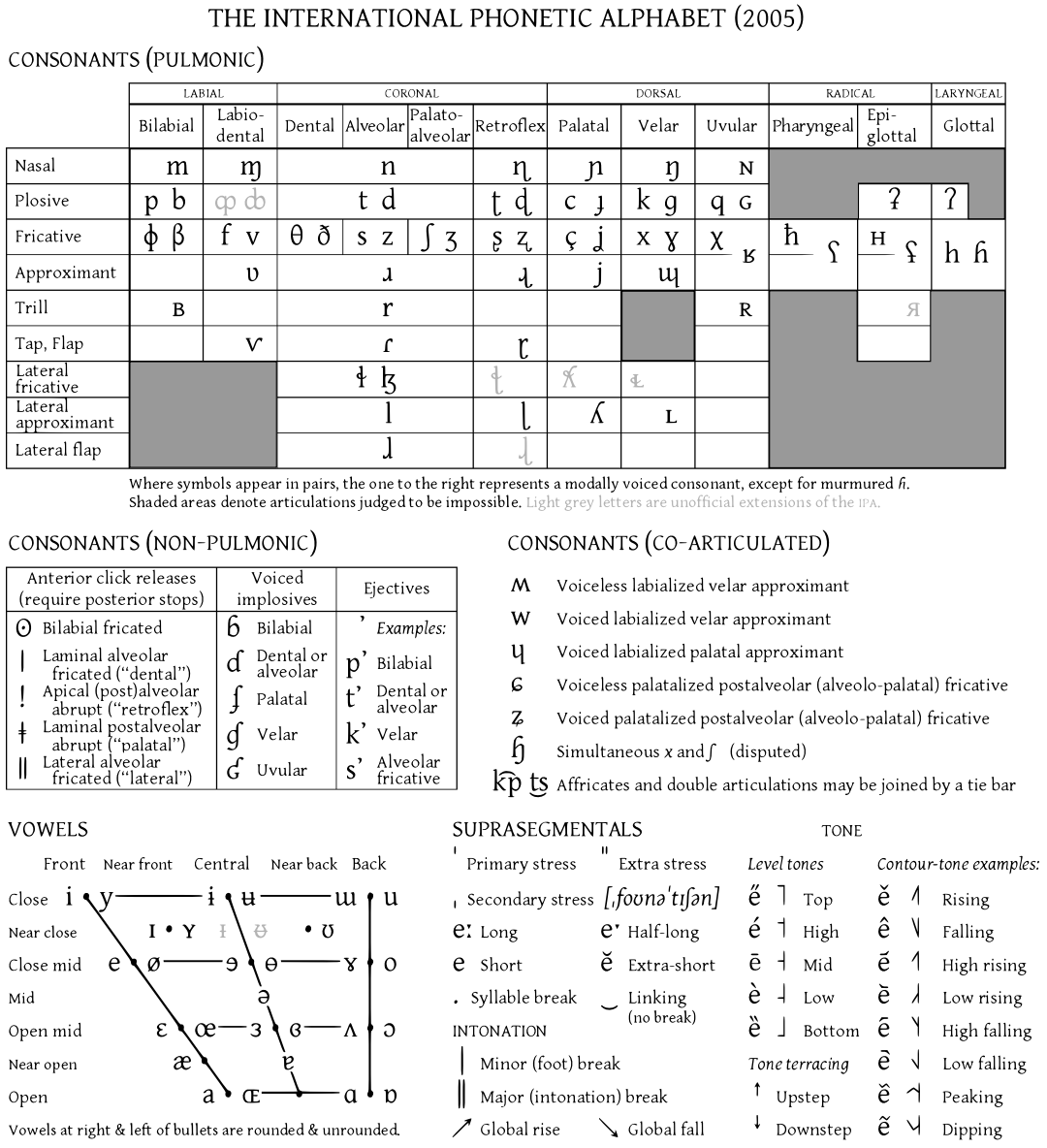


Figure A.1: The full international phonetic alphabet (IPA) as of 2005, including the labiodental flap and (in grey) some ad hoc symbols found in the literature (from wikipedia: http://en.wikipedia.org/wiki/File:IPA_chart_2005.png).

Bibliography

O. Anderson, P. Dalsgaard, and W. Barry. On the use of data-driven clustering technique for identification of poly- and mono-phonemes for four European languages. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume i, pages 121–124, 1994.

G. Aradilla. *Acoustic Models for Posterior Features in Speech Recognition*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2008.

G. Aradilla, H. Bourlard, and Magimai-Doss. Posterior features applied to speech recognition tasks with user-defined vocabulary. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3809–3812, 2009.

ArpaBet. CMU pronouncing dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict, Feb 2013.

E. Barnard, M. Davel, and C. van Heerden. ASR corpus design for resource-scarce languages. In *Proceedings of Interspeech*, pages 2847–2850, 2009.

J. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, International Computer Science Institute, Berkeley, April 1998.

M. Bisani and H. Ney. Bootstrap estimates for confidence intervals in ASR performance evaluation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 409–412, 2004.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

V. Blažek. On the internal classification of Indo-European languages: survey. http://www.phil.muni.cz/linguistica/art/blazek/bla-003.pdf, 2005.

H. Bourlard, N. Morgan, C. Wooters, and S. Renals. CDNN: a context dependent neural network for continuous speech recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 349–352, 1992.

G. Bouselmi, D. Fohr, and I. Illina. Multi-accent and accent-independent non-native speech recognition. In *Proceedings of Interspeech*, pages 2703–2706, 2008.

G. Bouselmi, D. Fohr, and I. Illina. Multilingual recognition of non-native speech using acoustic model transformation and pronunciation modeling. *International Journal of Speech Technology*, pages 1–11, 2012.

L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, D. Povey, A. Rastrow, R. C. Rose, and S. Thomas. Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4334–4337, 2010.

W. Byrne, P. Beyerlein, J.M. Huerta, S. Khudanpur, B. Marthi, J. Morgan, N. Peterek, J. Picone, D. Vergyri, and T. Wang. Towards language independent acoustic modeling. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 1029–1032, 2000.

H. Caesar. Integrating language identification to improve multilingual speech recognition. Technical Report Idiap-RR-24-2012, Idiap Research Institute, July 2012. http://publications.idiap.ch/downloads/reports/2012/Caesar_Idiap-RR-24-2012.pdf.

G. Chollet, F. T. Johansen, B. Lindberg, and F. Senia. LE2-4001 deliverable identification. Technical Report LE2-4001-SD1.3.4, ENST, Telenor, CPK and CSELT, 08 1998. http://www.speechdat.org/speechdat/deliverables/public/SD113V33.PDF.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991. ISBN 0471062596.

G.E. Dahl, Dong Yu, Li Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

M. Davel and O. Martirosian. Pronunciation dictionary development in resource-scarce environments. In *Proceedings of Interspeech*, pages 2851–2854, 2009.

S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Audio, Speech, and Language Processing*, 28(4):357–366, 1980.

M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle. Template-based continuous speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1377–1390, 2007.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society, series B*, 39(1):1–38, 1977.

S. Dupont, C. Ris, O. Deroo, and S. Poitoux. Feature extraction and acoustic modeling: an approach for improved generalization across languages and accents. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 29–34, 2005.

H. Engelbrecht and T. Schultz. Rapid development of an afrikaans-english speech-to-speech translator. In *Proceedings of International Workshop of Spoken Language Translation (IWSLT)*, 2005.

C. Fugen, S. Stuker, Hagen Soltau, F. Metze, and T. Schultz. Efficient handling of multilingual language models. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 441–446, 2003.

M.J.F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12(2):75 – 98, 1998.

P. N. Garner and D. Imseng. Statistical models for HMM/ANN hybrids. Technical Report Idiap-RR-11-2013, Idiap Research Institute, April 2013. http://publications.idiap.ch/downloads/reports/2013/Garner_Idiap-RR-11-2013.pdf.

J.-L. Gauvain and C.-H. Lee. Speaker adaptation based on MAP estimation of HMM parameters. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 558–561, 1993.

R. Gemello, F. Mana, and S. Scanzio. Experiments on HIWIRE database using denoising and adaptation with a hybrid HMM-ANN model. In *Proceedings of Interspeech*, pages 2429–2432, 2007.

B. Gold and N. Morgan. *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley & Sons, Inc, 2000.

S. Goronzy, S. Rapp, and R. Kompe. Generating non-native pronunciation variants for lexicon adaptation. *Speech Communication*, 42(1):109–123, 2004.

F. Grézl, M. Karafiát, and M. Janda. Study of probabilistic and bottle-neck features in multilingual environment. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 359–364, 2011.

A. Grichting. *Wallissertitschi Weerter*. Radio Rottu Oberwallis und Walliser Bote, 2011. ISBN 978-3-907816-74-5.

W. Heeringa and F. de Wet. The origin of Afrikaans pronunciation: a comparison to west Germanic languages and Dutch dialects. In *Proceedings of the Conference of the Pattern Recognition Association of South Africa*, pages 159–164, 2008. www.let.rug.nl/heeringa/dialectology/papers/prasa08.pdf.

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

H. Hermansky, D. P. W. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1635–1638, 2000.

X. D. Huang and M. A. Jack. Semi-continuous hidden Markov models for speech signals. *Computer Speech and Language*, 3(3):239–251, 1989.

B. Imperl, Z. Kacic, B. Horvat, and A. Zgank. Agglomerative vs. tree-based clustering for the definition of multilingual set of triphones. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 1273–1276, 2000.

D. Imseng and H. Bourlard. Speaker adaptive Kullback–Leibler divergence based hidden Markov models. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

D. Imseng, H. Bourlard, and M. Magimai.-Doss. Towards mixed language speech recognition systems. In *Proceedings of Interspeech*, pages 278–281, 2010.

D. Imseng, H. Bourlard, H. Caesar, P. N. Garner, G. Lecorvé, and A. Nanchen. MediaParl: Bilingual mixed language accented speech database. In *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT)*, pages 263–268, 2012a.

D. Imseng, H. Bourlard, and P. N. Garner. Using KL-divergence and multilingual information to improve ASR for under-resourced languages. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4869–4872, 2012b.

D. Imseng, H. Bourlard, and P. N. Garner. Boosting under-resourced speech recognizers by exploiting out of language data - case study on Afrikaans. In *Proceedings of the 3rd International Workshop on Spoken Languages Technologies for Under-resourced Languages*, pages 60–67, 2012c.

D. Imseng, J. Dines, P. Motlicek, P. N. Garner, and H. Bourlard. Comparing different acoustic modeling techniques for multilingual boosting. In *Proceedings of Interspeech*, 2012d.

D. Imseng, H. Bourlard, J. Dines, P. N. Garner, and M. Magimai.-Doss. Applying multi- and cross-lingual stochastic phone space transformations to non-native speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 2013a. to be published.

D. Imseng, P. Motlicek, H. Bourlard, and P. N. Garner. Using out-of-language data to improve an under-resourced speech recognizer. *Speech Communication*, 2013b. ISSN 0167-6393. doi: 10.1016/j.specom.2013.01.007.

IPA. The international phonetic association. http://www.langsci.ucl.ac.uk/ipa/, Feb 2013.

D. Johnson. ICSI quicknet software package. http://www.icsi.berkeley.edu/Speech/qn.html, 2004.

P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the 10th Machine Translation Summit*, pages 79–86, 2005.

J. Köhler. Language adaptation of multilingual phone models for vocabulary independent speech recognition tasks. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 417–420, 1998.

J. Köhler. Multilingual phone models for vocabulary-independent speech recognition tasks. *Speech Communication*, 35:21–30, 2001.

S. Kullback. The Kullback-Leibler distance. *The American Statistician*, 41(4):340–341, in Letters to the Editor, November 1987.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. http://projecteuclid.org/DPubS/Repository/1.0/Disseminate?view=body&id=pdf_1&handle=euclid.aoms/1177729694.

C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2):171–185, 1995.

Lwazi. A telephone-based speech-driven information system for South Africa. http://www.meraka.org.za/lwazi/pdf/lwazibrochure.pdf, March 2013.

N. Morgan and H. Bourlard. Continuous speech recognition using multilayer perceptrons with hidden markov models. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 413–416, 1990.

N. Morgan and H. Bourlard. Continuous speech recognition: an introduction to the hybrid HMM/connectionist approach. *IEEE Signal Processing Magazine*, 12(3):24–42, May 1995.

T. Niesler. Language-dependent state clustering for multilingual acoustic modelling. *Speech Communication*, 49:453–463, 2007.

N. Nocerino, F. Soong, L. Rabiner, and D. Klatt. Comparative study of several distortion measures for speech recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 10, pages 25–28, 1985.

J. Novak, N. Minematsu, K. Hirose, C. Hori, H. Kashioka, and P. Dixon. Improving WFST-based G2P conversion with alignment constraints and RNNLM N-best rescoring. In *Proceedings of Interspeech*, 2012.

N. Oostdijk. The spoken Dutch corpus. Overview and first evaluation. In *In Proceedings of the Second International Conference on Language Resources and Evaluation*, volume II, pages 887–894, 2000.

J. P. Openshaw, Z. P. Sun, and J.S. Mason. A comparison of composite features under degraded speech in speaker recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 371–374, 1993.

G. Perennou. B.D.L.E.X. : A data and cognition base of spoken French. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 11, pages 325–328, 1986.

D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Karafiàt, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas. Subspace gaussian mixture models for speech recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4330–4333, 2010.

D. Povey, M. Karafiát, A. Ghoshal, and P. Schwarz. A symmetrization of the subspace Gaussian mixture model. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4504–4507, 2011.

J. Psutka, L. Müller, and J. V. Psutka. Comparison of MFCC and PLP parameterization in the speaker independent continuous speech recognition task. In *Proceedings of Eurospeech*, pages 1813–1816, 2001.

Y. Qian, J. Xu, D. Povey, and J. Liu. Strategies for using MLP based features with limited target-language training data. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 354–358, 2011.

M. Raab, R. Gruhn, and E. Nöth. Multilingual weighted codebooks for non-native speech recognition. In *Proceedings of the 11th International Conference on Text, Speech and Dialogue*, pages 485–492, 2008.

L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, April 1993. ISBN 0130151572.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Ramya Rasipuram and Mathew Magimai.-Doss. Acoustic data-driven grapheme-to-phoneme conversion using KL-HMM. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4841–4844, 2012.

M. D. Richard and R. P. Lippmann. Neural network classifiers estimate Bayesian a posterior probabilities. *Neural Computation*, 3:461–483, 1991.

G. Rigoll and D. Willett. A NN/HMM hybrid for continuous speech recognition with a discriminant nonlinear feature extraction. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 9–12, 1998.

I Rogina. Automatic architecture design by likelihood-based context clustering with crossvalidation. In *Proceedings of Eurospeech*, volume 97, pages 1223–1226, 1997.

J. Rottland and G. Rigoll. Tied posteriors: an approach for effective introduction of context dependency in hybrid NN/HMM LVCSR. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 1241–1244, 2000.

S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana. On the use of a multilingual neural network front-end. In *Proceedings of Interspeech*, pages 2711–2714, 2008.

F. Schiel. Aussprache-lexikon PHONOLEX. http://www.phonetik.uni-muenchen.de/forschung/Bas/BasPHONOLEXeng.html, March 2013.

T. Schultz. Multilingual acoustic modeling. In *Multilingual Speech Processing*, pages 71–122. Academic Press, 2006. ISBN 978-0-12-088501-5.

T. Schultz and A. Waibel. Polyphone decision tree specialization for language adaptation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1707–1710, 2000.

T. Schultz and A. Waibel. Language independent and language adaptive acoustic modeling for speech recognition. *Speech Communication*, 35:31–51, 2001.

J. C. Segura, T. Ehrette, A. Potamianos, D. Fohr, I. Illina, P-A. Breton, V. Clot, R. Gemello, M. Matassoni, and P. Maragos. The HIWIRE database, a noisy and non-native English speech corpus for cockpit communication. http://cvsp.cs.ntua.gr/projects/pub/HIWIRE/WebHome/HIWIRE_db_description_paper.pdf, 2007.

K. Shinoda and T. Watanabe. Acoustic modeling based on the MDL principle for speech recognition. In *Proceedings of Eurospeech*, pages 99–102, 1997.

K. C. Sim. Discriminative product-of-expert acoustic mapping for cross-lingual phone recognition. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 546–551, 2009.

K. C. Sim and H. Li. Robust phone set mapping using decision tree clustering for cross-lingual phone recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4309–4312, 2008.

S. Soldo, M. Magimai.-Doss, J. P. Pinto, and H. Bourlard. Posterior features for template-based ASR. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4864–4867, 2011.

A. Stolcke, F. Grezl, M.-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri. Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 321–324, 2006.

L. Tòth, J. Frankel, G. Gosztolya, and S. King. Cross-lingual portability of MLP-based tandem features - a case study for English and Hungarian. In *Proceedings of Interspeech*, pages 2695–2698, 2008.

N. Ueda, R. Nakano, Z. Ghahramani, and G.E. Hinton. Split and merge EM algorithm for improving Gaussian mixture density estimates. In *Proceedings of the 1998 IEEE Signal Processing Society Workshop*, pages 274–283, 1998.

D. Van Compernolle. Recognizing speech of goats, wolves, sheep and...non-natives. *Speech Communication*, 35:71–79, 2001.

C. van Heerden, E. Barnard, and M. Davel. Basic speech recognition for spoken dialogues. In *Proceedings of Interspeech*, pages 3003–3006, 2009.

N. T. Vu, F. Kraus, and T. Schultz. Multilingual a-stabil: A new confidence score for multilingual unsupervised training. In *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT)*, pages 183–188, 2010.

N.T. Vu, F. Kraus, and T. Schultz. Rapid building of an asr system for under-resourced languages based on multilingual unsupervised training. In *Proceedings of Interspeech*, pages 3145–3148, 2011.

Z. Wang, T. Schultz, and A. Waibel. Comparison of acoustic model adaptation techniques on non-native speech. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 540–543, 2003.

T. Ward, S. Roukos, C. Neti, J. Gros, M. Epstein, and S. Dharanipragada. Towards speech understanding across multiple languages. In *The 5th International Conference on Spoken Language Processing*, 1998.

J.C. Wells. SAMPA computer readable phonetic alphabet. http://www.phon.ucl.ac.uk/home/sampa/, Feb 2013.

F. Weng, H. Bratt, L. Neumeyer, and A. Stolcke. A study of multilingual speech recognition. In *Proceedings of Eurospeech*, pages 359–362, 1997.

S. J. Young, J. J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the workshop on Human Language Technology*, pages 307–312, 1994.

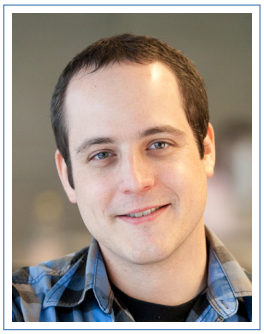
S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK, 2006.

H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A.W. Black, and K. Tokuda. The HMM-based speech synthesis system version 2.0. In *Proceedings of the sixth ISCA Workshop on Speech Synthesis (ISCA SSW6)*, pages 294–299, 2007. http://hts.sp.nitech.ac.jp/.

Index

David Imseng

+41 79 407 64 15
+41 27 721 77 76
dimseng@idiap.ch
www.idiap.ch/~dimseng



Current

Since 2009 **Research assistant (Ph.D. student)**, *Idiap Research Institute*, Martigny, Switzerland.

Working in the general field of multilingual speech recognition with focus on neural network based multilingual acoustic modeling approaches.

Education

2009–2013 **Ph.D. in Electrical Engineering**, *Ecole Polytechnique Fédérale (EPFL)*, Lausanne, Switzerland.

- Thesis Topic: *Multilingual speech recognition – a posterior based approach*
- Supervisor: Prof. Hervé Bourlard

2006–2009 **M.Sc. in Communication Systems**, *EPFL*, Lausanne, Switzerland.

- Thesis Topic: *Novel initialization methods for speaker diarization*
- Supervisors: Dr. Gerald Friedland and Prof. Hervé Bourlard

2003–2006 **B.Sc. in Communication Systems**, *EPFL*, Lausanne, Switzerland.

Experience

Sept. 2012 – Dec. 2012 **Visiting scholar**, *International Computer Science Institute (ICSI)*, Berkeley, California USA.

Worked on the IARPA Babel Program that is focused on building speech recognition solutions with self-imposed time and data limitations for a variety of languages.

Sept. 2008 – March 2009 **Visiting scholar**, *International Computer Science Institute (ICSI)*, Berkeley, California USA.

Worked on novel initialization methods for speaker diarization and contributed to the diarization system of ICSI that participated in the National Institute of Standards and Technology Rich Transcription 2009 Evaluation.

Oct. 2007 – Feb. 2008 **Semester project**, *Ecole Polytechnique Fédérale (EPFL)*, Lausanne, Switzerland.

Worked on recognition of vowels using artificial neural networks in the audiovisual communications lab under the supervision of Prof. Martin Vetterli.

March 2007 – Aug. 2007 **Internship**, *IMRA EUROPE S.A.S.*, Sophia Antipolis, France.
Design of a Graphical User Interface (Microsoft Foundation Classes) and development of an adaptive pattern recognition algorithm (Matlab).

March 2006 – Sept. 2006 **Semester Project and Internship**, *Idiap Research Institute*, Martigny, Switzerland.
Worked on automatic alignment of sentence transcripts with associated speech utterances under the supervision of Prof. Hervé Bourlard.

Aug. 2005 – Sept. 2005 **Internship**, *Swisscom*, Ostermundigen, Switzerland.
Migration of the intranet appearance (team of five students).

Professional activities

Reviewer for ACM Transactions on Speech and Language Processing (since 2012)

Reviewer for IEEE Transactions on Audio Speech and Language Processing (since 2011)

Student supervision

Computer Skills

- Linux, Microsoft Windows, Mac OS X
- Bash, C, C++, Java, HTML, XML, Python, Ruby
- Latex, Matlab, Microsoft Office, Hidden Markov Model Toolkit (HTK)

Languages

German	native
English	fluent
French	fluent
Italian	basic

Honors

- Idiap PhD Student Research Award 2012: for outstanding publication records and for excellence in the research topic *Multilingual Speech Recognition*

Interests and Activities

- Playing the fife (playing in a fife and drum corps, referee at competitions, member of the regional and federal technical committee)
- Squash, Soccer, Floorball, Snowboarding, Hiking

Publications

In journals

[1] **David Imseng**, Hervé Bourlard, John Dines, Philip N. Garner, and Mathew Magimai.-Doss. Applying multi- and cross-lingual stochastic phone space transformations to non-native speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 2013.

[2] **David Imseng**, Petr Motlicek, Hervé Bourlard, and Philip N. Garner. Using out-of-language data to improve an under-resourced speech recognizer. *Speech communication*, 2013.

[3] Gerald Friedland, Adam Janin, **David Imseng**, Xavier Anguera, Luke Gottlieb, Marijn Huijbregts, Mary Tai Knox, and Oriol Vinyals. The ICSI RT-09 speaker diarization system. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):371–381, 2012.

[4] Hervé Bourlard, John Dines, Mathew Magimai.-Doss, Philip N. Garner, **David Imseng**, Petr Motlicek, Hui Liang, Lakshmi Saheer, and Fabio Valente. Current trends in multilingual speech processing. *Sadhana*, 36(5):885–915, 2011.

[5] **David Imseng** and Gerald Friedland. Tuning-robust initialization methods for speaker diarization. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2028–2037, 2010.

In conference proceedings

[1] **David Imseng** and Hervé Bourlard. Speaker adaptive kullback-leibler divergence based hidden markov models. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[2] **David Imseng**, Hervé Bourlard, Holger Caesar, Philip N. Garner, Gwénolé Lecorvé, and Alexandre Nanchen. Mediaparl: Bilingual mixed language accented speech database. In *Proceedings of the 2012 IEEE Workshop on Spoken Language Technology*, 2012.

[3] **David Imseng**, John Dines, Petr Motlicek, Philip N. Garner, and Hervé Bourlard. Comparing different acoustic modeling techniques for multilingual boosting. In *Proceedings of Interspeech*, 2012.

[4] Milos Cernak, **David Imseng**, and Hervé Bourlard. Robust triphone mapping for acoustic modeling. In *Proceedings of Interspeech*, 2012.

[5] **David Imseng**, Hervé Bourlard, and Philip N. Garner. Boosting under-resourced speech recognizers by exploiting out of language data - case study on Afrikaans. In *Proceedings of the 3rd workshop on Spoken Language Technologies for Under-resourced languages*, pages 60–67, 2012.

[6] **David Imseng**, Hervé Bourlard, and Philip N. Garner. Using KL-divergence and multilingual information to improve ASR for under-resourced languages. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4869–4872, 2012.

[7] **David Imseng**, Ramya Rasipuram, and Mathew Magimai.-Doss. Fast and flexible Kullback-Leibler divergence based acoustic modeling for non-native speech recognition. In *Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding*, pages 348–353, 2011.

[8] **David Imseng**, Hervé Bourlard, John Dines, Philip N. Garner, and Mathew Magimai.-Doss. Improving non-native ASR through stochastic multilingual phoneme space transformations. In *Proceedings of Interspeech*, pages 537–540, 2011.

[9] **David Imseng**, Hervé Bourlard, Mathew Magimai.-Doss, and John Dines. Language dependent universal phoneme posterior estimation for mixed language speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5012–5015, 2011.

[10] **David Imseng**, Mathew Magimai.-Doss, and Hervé Bourlard. Hierarchical multilayer perceptron based language identification. In *Proceedings of Interspeech*, pages 2722–2725, 2010.

[11] **David Imseng**, Hervé Bourlard, and Mathew Magimai.-Doss. Towards mixed language speech recognition systems. In *Proceedings of Interspeech*, pages 278–281, 2010.

[12] Andreas Stolcke, Gerald Friedland, and **David Imseng**. Leveraging speaker diarization for meeting recognition from distant microphones. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4390–4393, 2010.

[13] **David Imseng** and Gerald Friedland. An adaptive initialization method for speaker diarization based on prosodic features. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4946–4949, 2010.

[14] **David Imseng** and Gerald Friedland. Robust speaker diarization for short speech recordings. In *Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding*, pages 432–437, 2009.

Research reports

[1] Philip N. Garner and **David Imseng**. Statistical models for HMM/ANN hybrids. Technical Report Idiap-RR-11-2013, Idiap Research Institute, April 2013. `http://publications.idiap.ch/downloads/reports/2013/Garner_Idiap-RR-11-2013.pdf`.

[2] Petr Motlicek, Philip N. Garner, **David Imseng**, and Fabio Valente. Application of subspace Gaussian mixture models in contrastive acoustic scenarios. Technical Report Idiap-RR-20-2012, Idiap Research Institute, July 2012. `http://publications.idiap.ch/downloads/reports/2012/Motlicek_Idiap-RR-20-2012.pdf`.

[3] **David Imseng** and John Dines. Decision tree clustering for KL-HMM. Technical Report Idiap-Com-01-2012, Idiap Research Institute, February 2012. `http://publications.idiap.ch/downloads/reports/2012/Imseng_Idiap-Com-01-2012.pdf`.

[4] **David Imseng**. Novel initialization methods for speaker diarization. Technical Report Idiap-RR-07-2009, Idiap Research Institute, May 2009. Master's thesis, `http://publications.idiap.ch/downloads/reports/2009/Imseng_Idiap-RR-07-2009.pdf`.