Dynamic Pronunciation Modeling using Phonetic Features and Symbolic Speaker Adaptation for Automatic Speech Recognition

Thèse présentée au
Département de Systèmes de Communication
Ecole Polytechnique Fédérale de Lausanne

pour l'obtention du grade de Docteur Es Sciences Techniques

par

Kyung-Tak LEE

Ingénieur en systèmes de communication, EPFL

Composition du jury:

Président:	Prof. Emre TELATAR	EPFL
Directeur de thèse:	Prof. Christian WELLEKENS	Institut Eurécom
Rapporteurs:	Dr. Ingunn AMDAL	Telenor R&D
	Dr. Andrzej DRYGAJLO	EPFL
	Prof. Jean-Pierre MARTENS	RUG-ELIS
	Prof. Dirk SLOCK	Institut Eurécom

Abstract

State-of-the-art speech recognition systems typically contain a single or very few phonetic transcriptions per lexical word. Such a sparse encoding of pronunciation variants is challenged by the great number of pronunciation differences that exist in reality. A standard method in pronunciation variation modeling consists of adding more pronunciation variants to the lexicon, but adding too many of them increases risks of lexical confusability and may also decrease recognition performance. A possible way to address this issue is to limit the number of variants to add based on a certain criterion. This approach has however the disadvantage of also limiting pronunciation coverage, which is an important factor for spontaneous and non-native speech.

This dissertation studies several methods that *dynamically* model pronunciation variation. In comparison to more traditional (static) methods, more pronunciations can be modeled, thus ensuring a higher coverage, but they are activated at different times during recognition so that lexical confusability can still be limited. Two aspects of this dynamic approach were investigated:

Level of modeling : dynamic pronunciation modeling was applied separately to the phonetic and acoustic levels, by respectively adapting the lexicon and acoustic models to the pronunciations of the speech utterance to recognize. These modifications were governed by the automatic extraction of *phonetic (articulatory) features* from the input speech.

Level of dynamism : dynamic pronunciation modeling was applied both on per utterance and per speaker bases: besides the utterance-level methods mentioned in the previous point, another technique based on *symbolic speaker adaptation* built a separate lexicon per speaker. The objective of the adaptation process was to create a profile per speaker that modeled the pronunciation characteristics of the latter by a combination of several pronunciation styles (called here "speech varieties", SV) known by the system. These profiles influenced how the canonical lexicon was expanded with pronunciation variants to yield a speaker-dependent lexicon. The method was evaluated with multiple dialects and foreign accents as the modeled SVs.

Recognition performance increased significantly when the dynamic lexicon was combined with canonical transcriptions, compared both to the baseline performance and to the result obtained with a lexicon augmented with new pronunciation variants, but not dynamically modified during recognition. On the other hand, although dynamic acoustic models showed comparable performance to the best result obtained with static models, improvement relative to the baseline was not significant, suggesting therefore further study in this area.

Automatic detection of phonetic features led to a smaller confusion distance between alternatives than phones, suggesting that features are more appropriate to generate pronunciation

variants. Besides, combination of phonetic features and a pronunciation model based on decision trees helped to significantly increase the phone accuracy of the system. The feature extraction method was evaluated on both read and spontaneous speech; only slight degradation per feature was observed when shifting to spontaneous speech, which however led to a substantial frame-level degradation when these single feature errors were combined.

Basic experiments with symbolic speaker adaptation only slightly improved performance. However, the technique performed better when combined with acoustic speaker adaptation (MLLR), suggesting that the two levels of modeling are complimentary. Furthermore, an analysis of intermediate results indicated that a lot of pronunciations selected by the system during the adaptation process were biased towards baseforms, even though there were clearly pronunciation variations between the evaluated SVs. Expansion of the phone inventory with more SV-inclusive phones significantly reduced this preference for baseforms and further increased word recognition performance in most cases. Some additional experiments suggested that symbolic speaker adaptation can target a speaker's speech variety with small adaptation data and that its SV-blending scheme is more efficient than a standard SV-classification scheme to model speakers from speech varieties unknown by the system.

Résumé

Les systèmes standards de reconnaissance de la parole contiennent typiquement une seule ou très peu de transcriptions phonétiques par mot lexical. Un nombre si faible en variantes de prononciation n'est pas suffisant pour modéliser toutes les différences de prononciation qui existent dans la réalité. Une méthode classique en modélisation des variations de prononciation consiste à ajouter des variantes de prononciation dans le lexique, mais en ajouter trop augmente les risques de confusion lexicale et peut aussi diminuer les performances en reconnaissance. Une voie possible pour résoudre ce problème est de limiter le nombre de variantes à ajouter sur la base d'un certain critère. Cette approche a cependant le désavantage de limiter également l'étendue de la modélisation des prononciations, un facteur important pour les paroles dites "spontanée" et "non-native".

Cette dissertation étudie plusieurs méthodes qui modélisent *dynamiquement* les variations de prononciation. En comparaison avec des méthodes plus traditionnelles (statiques), plus de prononciations peuvent être modélisées, ce qui assure ainsi une plus grande étendue de la modélisation, mais elles sont activées à différents moments durant la reconnaissance pour que la confusion lexicale puisse toujours être limitée. Deux aspects de cette approche dynamique ont été examinés:

Niveau de la modélisation : la modélisation dynamique des prononciations a été appliquée séparément aux niveaux phonétique et acoustique, en adaptant respectivement le lexique et les modèles acoustiques aux prononciations contenues dans la parole à reconnaître. Ces modifications ont été guidées par l'extraction automatique de *traits phonétiques (articulatoires)* contenues dans la parole.

Niveau du dynamisme : la modélisation dynamique des prononciations a été appliquée aussi bien au niveau du locuteur qu'au niveau de la phrase: en plus des méthodes "par phrase" mentionnées dans le point précédent, une autre technique basée sur une *adaptation symbolique au locuteur* a construit un lexique spécifique pour chaque locuteur. L'objectif de l'adaptation était de créer un profil par locuteur qui modélise les caractéristiques de prononciation de celui-ci par une combinaison de plusieurs styles de prononciation (appelés ici "speech varieties" en anglais) connus par le système de reconnaissance. Ces profils ont influencé la façon d'augmenter la taille du lexique canonique avec des variantes de prononciation afin d'obtenir un lexique adapté au locuteur. La méthode a été évaluée en utilisant plusieurs dialectes et accents étrangers comme styles de prononciation.

La performance en reconnaissance a été significativement améliorée lorsque le lexique dynamique a été combiné avec les transcriptions canoniques, en comparaison avec la performance de base et le résultat obtenu avec un lexique augmenté avec de nouvelles variantes de prononciation, mais non modifié dynamiquement pendant la phase de reconnaissance. Par contre, bien

que les modèles acoustiques dynamiques ont obtenu des performances comparables au meilleur résultat obtenu avec des modèles statiques, l'amélioration par rapport à la performance de base n'a pas été significative et requiert une étude plus approfondie dans ce domaine.

La détection automatique des traits phonétiques a conduit à une distance de confusion plus faible entre les résultats alternatifs qu'avec les phones, ce qui suggère que ces traits sont plus appropriés pour générer des réseaux de prononciation. En outre, la combinaison des traits phonétiques avec un modèle de prononciation basé sur des arbres de décision a permis d'améliorer de façon significative le taux de reconnaissance en phonème du système. La méthode d'extraction des traits a été évaluée aussi bien sur de la parole "spontanée" que sur de la parole "lue"; une faible dégradation par trait a été perceptible lorsqu'on est passé à de la parole spontanée, mais cela a tout de même entraîné une dégradation importante au niveau des trames lors de la combinaison de ces traits.

Les expériences de base sur l'adaptation symbolique au locuteur n'ont donné que peu d'amélioration de la performance. Par contre, la technique a donné de meilleurs résultats lorsqu'elle a été combinée avec une méthode d'adaptation acoustique au locuteur (MLLR en anglais), suggérant ainsi que les deux niveaux de modélisation sont complémentaires. En outre, une analyse des résultats intermédiaires a indiqué que beaucoup de prononciations sélectionnées par le système pendant la phase d'adaptation favorisaient les prononciations de base, bien qu'il y avait clairement des variations de prononciation entre les différents styles. Une augmentation du nombre de phonèmes plus spécifiques aux dialectes et accents étrangers a permis de réduire significativement ce biais et d'améliorer davantage les performances en reconnaissance de mots dans la plupart des cas. Des expériences supplémentaires ont suggéré que la reconnaissance symbolique au locuteur peut identifier le style de prononciation d'un locuteur avec très peu de données d'adaptation, et que son principe de combinaison des styles est plus efficace que le principe standard de classification pour modéliser des locuteurs avec des styles de prononciation qui ne sont pas connus par le système.

Acknowledgments

First and foremost, I thank God for having given me the strength and courage to accomplish this dissertation. Nothing would have been possible without Him.

My next thanks naturally go to my family. Their unconditional love and support throughout this work were far more helpful than any lesson I received or any publication I read.

I would like to express my sincere gratitude to my thesis supervisor, Christian Wellekens, because he guided me and provided me with unfailing support throughout my thesis. He especially never lost his patience in answering all my questions (and God knows how many I had).

During the course of my thesis, I had the privilege of working almost the entire year 2002 at Motorola Labs in Schaumburg (USA). I am indebted to all the people who contributed in making this internship possible. First, Sreeram Balakrishnan and Pierre Demartines, two of my former supervisors during a previous assignment, who kindly took care of all the necessary steps until the period of my relocation. Then, Carla Mote and Barbara Moskal, who efficiently took in charge of my relocation and made themselves available for any help during and even after my assignment. Finally and especially, I am grateful to Steven Nowlan and Yan Ming Cheng, my industrial supervisors, who hosted me during that period and shared with me their incredible experience about research in speech and its implication in the industrial world.

My stay at Motorola Labs would have been very difficult without the help of many smart people of the lab and I thank them all for their great support. Three of them that I would like to thank in particular are Lynette Melnar, for her invaluable expertise in linguistics and her availability, Jim Talley, for his great advice in pronunciation modeling and his concern about doing things well and in minute detail, and Dušan Macho, for being an excellent discussion partner in many domains.

I also would like to thank the members of the Ph.D. examining board: Ingunn Amdal, Andrzej Drygajlo, Jean-Pierre Martens, Dirk Slock and Emre Telatar. They all promptly and kindly accepted to review my dissertation despite their heavy responsabilities and tight schedules in their respective domains. I also greatly appreciate the help of Simon King and Qian Yang, who provided me with useful information during the course of my thesis.

Last but surely not the least, my Ph.D. years would not have been very colorful without the presence of many friends. Particular thanks to Janet Cahn, Frédéric Chifflet, Dušan & Merce Macho, Marc Monserrat, Phuc-Tri Nguyen, Céline Steenkeste and Geoffrey Wilfart. They all created an enjoyable atmosphere and helped me to remind that work is important, but is not everything in life.

Contents

List of Figures

List of Tables

List of abbreviations

ANN - Artifical Neural Network
As - Asian(-accented English)
ASA - Acoustic Speaker Adaptation
ASR - Automatic Speech Recognition
ASS - Automatic Speech Synthesis
ASU - Automatic Speech Understanding
BEEP - British English Example Pronunciation (dictionary)
Br - British (English)
CART - Classification And Regression Trees
CAT - Cluster Adaptive Training
CMU - Carnegie Mellon University (dictionary)
CSR - Continuous Speech Recognition
DBN - Dynamic Bayesian Network
DP - Dynamic Programming
EMA - Electro-Magnetic Articulograph
GRD - Gelfand Ravishankar Delp
HAMM - Hidden Articulator Markov Model
HMM - Hidden Markov Model
HTK - Hidden markov model ToolKit
ICSLP - International Conference on Spoken Language Processing
In - Indian (English dialect)
IPA - International Phonetic Alphabet
ISCA - International Speech Communication Association
ITRW - ISCA Tutorial and Research Workshop
LDA - Linear Discriminant Analysis
LDM - Linear Dynamic Model
LM - Language Model
MAP - Maximum A Posteriori
MFCC - Mel-Frequency Cepstral Coefficient
ML - Maximum Likelihood
MLLR - Maximum Likelihood Linear Regression
MR - Matching Ratio
MV - Multi-Valued
NI - Northern Inland (English dialect)
NICO - Neural Inference COmputation (ANN toolkit)
NIST - National Institute of Standards and Technology
OOV - Out-Of-Vocabulary
PCA - Principal Component Analysis
PCPM - Partial Change Phone Model
PER - Phone Error Rate

PMLA	-	Pronunciation Modeling and Lexicon Adaptation
SA	-	(dialect sentence of TIMIT)
SAE	-	Standard American English
SD	-	Speaker-Dependent
SI	-	(phonetically-diverse sentence of TIMIT)
SLPM	-	State-Level Pronunciation Modeling
SPE	-	Sound Pattern of English
SSA	-	Symbolic Speaker Adaptation
SV	-	Speech Variety
SVP	-	Speech Variety Profile
SX	-	(phonetically-compact sentence of TIMIT)
TIMIT	-	Texas Instruments - Massachusetts Institute of Technology (speech database)
WA	-	Word Accuracy
WCR	-	Word Correct Rate
WER	-	Word Error Rate
WFST	-	Weighted Finite State Transducer
WSJ0	-	Wall Street Journal (speech database)
WSJCAM0	-	Wall Street Journal CAMbridge (speech database)
XRMB	-	X-Ray MicroBeam

Chapter 1

Introduction

"Speech technology is destined to play a decisive role in this societal transformation by virtue of its ability to facilitate and automate communication between humans and machines. (...) Cellular phones, personal digital assistants and computers, "smart" chips in the home, car and office will all make extensive use of speech technology." Greenberg wrote in [57]. Who has indeed never seen in TV-series or movies, or at least imagined, cars that talk and understand whatever we say and can for instance drive us safely from a point A to a point B ? Or robots that can communicate with us like a true human being ?

Although such technology is still too early to be conceived in real life, progresses are being made towards that goal. A common aspect between the two examples (cars and robots) mentioned above is that they speak the same language as us, so that we do not need to put more effort in talking to machines than to humans. In order to make this perspective a reality, machines must first be capable of identifying any sequence of words that humans say; this is the concept of *speech recognition*.

At this point, someone may notice: "But speech recognition technology already exists !". It does exist indeed, but in a form simplified in a way or another. For example, possibilities of using vocal commands implemented nowadays in mobile phones are often limited to single words or to some easy recognition tasks with small vocabulary (*e.g.*, sequence of digits). Alternatively, some mobile phones can recognize a sequence of arbitrary words, but under the condition that exactly the same sequence was recorded in a previous session. Dictation systems do allow the recognition of continuous speech, but again with some constraints. Ian Stobie, a journalist who used such a dictation system, reported in his article [133]: "you need a powerful PC and quiet surroundings to have much chance of success. (...) Initial training takes about 10 minutes, but for accuracy to improve you must take the time to correct the errors it makes as you go along. I've probably spent five hours in total in training it, but it now knows the idiosyncrasies of my pronunciation and vocabulary.". The good point about this is that, in the end, the dictation system worked well for him, but after what amount of effort ? There are some exceptions that require less effort and even let speakers utter in conversational style, but most often when vocabulary and grammar are tuned for a specific task (*e.g.*, radiology).

All these examples of restrictions suggest that speech recognition systems need to be more robust against various factors. There are three main factors in speech technology that need to be taken into account, the task, the speaker and the environment:

The task : it corresponds to the "topic" on which the recognition is performed. A specific

task (*e.g.*, phone number dialing) generally requires a small amount of vocabulary words and/or combinations of these words. The system can perform fairly well in those situations because sequences to recognize are quite predictable. On the other hand, when a task has a more general scope and implies more words and possible word combinations, things are more complicated and recognition performance is sensitively lower. Much research is therefore intended for improving recognition accuracy with large vocabulary and more complex grammars.

The speaker : two speakers do not pronounce words in the same way, and even the pronunciation of a single speaker varies in time. Many factors are responsible for this pronunciation variation, for instance age, gender or emotional state. Most standard speech recognition systems simplify however the problem by assuming only a few possible phonetic transcriptions per word (even a single transcription in some cases), which is not enough to cover all possible pronunciations. This is especially true in conversational (also called "spontaneous") speech, which includes a lot more variations than in carefully read speech. The situation gets even worse when a person speaks with a certain dialect or foreign accent. All these pronunciation variations not accounted for by the system provoke a substantial drop in performance.

The environment : recognition systems are much sensitive to noise due to the location where the recognition is performed (background noise) and to the equipment used (channel noise). Even a change of microphone between the moment a system is trained and the moment it is evaluated can substantially affect the performance. Background noise also affects pronunciation variation and makes therefore the situation worse, because people tend to hyperarticulate in adverse conditions in order to be better understood (this is called the "Lombard effect"). A comprehensive study of methods against noise can be found for instance in Junqua and Haton [75].

1.1 Brief description of pronunciation modeling

The general goal of this dissertation is to better account for characteristics inherent to a speaker and differences between speakers by modeling pronunciation variation. Research in this field is far from being new since the first contributions were already noticed in the early 1970's. However, this topic has especially become important in the past ten years due to the availability of spontaneous speech databases (*e.g.*, Switchboard [54]), which contain a lot more pronunciation variation than in read speech. The gain of interest in this field is testified among others by the two workshops on pronunciation modeling organized during the past five years (1998 and 2002).

The basic idea of pronunciation modeling consists of explicitly including more alternative pronunciations into the speech recognition system, typically inside a component called the *lexicon* (cf. section 2.3.3 for more information). However, early works showed that it is not enough to just take account of all possible pronunciations, because a lot of them are similar to each other but correspond to different words. Consequently, they may increase the number of possible confusions and lead to a drop in recognition performance; this is a problem called *lexical confusability* (cf. section 3.6.1). Many contributions aim therefore at reducing this risk of confusion, in most cases by keeping only the most representative variants based on a certain criterion.

A detailed description about pronunciation variation modeling will be given in chapter 3.

1.2 Objectives of this dissertation

Although a careful selection of pronunciation variants can limit the risks of lexical confusability, it also limits pronunciation coverage, which is an important factor to consider when dealing with spontaneous and non-native speech. This is a possible reason why most contributions in this field led to only small improvements. An experiment made by McAllaster et al. [104] on simulated data revealed however that when pronunciations found in the lexicon exactly matched those found in the data, recognition performance increased by five to ten times. It is therefore important to insure good pronunciation coverage while still reducing lexical confusability.

Following this line of thought, this dissertation will study the possibilities of *dynamically* modeling pronunciation variation: it consists of keeping more pronunciation variants to insure a better pronunciation coverage, but of activating them at different times *during recognition* to reduce lexical confusability. The few existing contributions in this area focused on a *soft* activation process, which only changes the relative importance of pronunciation variants through probabilities but without rejecting any of them. In contrast with previous works, this dissertation will rather focus on a *hard* process, which consists of keeping a subset of the variants and of eliminating the rest. Two aspects of dynamic pronunciation modeling will be investigated:

Level of modeling : this dissertation will first study the dynamic approach at the lexicon level, by selecting during recognition a set of pronunciation variants per utterance. The selection process will be guided by the automatic extraction of *phonetic features* - which describe how a segment of speech is produced in terms of the human articulatory system - from the input speech. Then, the study will be extended to a lower (acoustic) level of the recognition system, based on a concept called *state-level pronunciation modeling*.

Level of dynamism : the two levels of modeling above were processed on a per utterance basis, that is, a separate lexicon or acoustic models were created for each utterance. Additionally, a speaker-level (pseudo-)dynamic pronunciation modeling was also investigated by introducing the concept of *symbolic speaker adaptation*. The issue of modeling multiple dialects and foreign accents will also be addressed through this method.

1.3 Outline of this dissertation

This dissertation is divided in the following chapters:

- Chapter 2 will review the basic concepts of automatic speech recognition (ASR). In particular, the most popular system based on Hidden Markov Models (HMM) will be described.
- Chapter 3 will review the basic concepts of pronunciation variation modeling. The problems associated with ASR systems due to this factor will be described as well as some methods proposed in the literature to address this issue.
- Chapter 4 will introduce the concept of dynamic pronunciation modeling at the lexicon level, based on the detection of phonetic features. A literature survey on the latter will first be given, then the proposed method and related experiments will be described in detail.

- Chapter 5 will extend the dynamic approach to the acoustic (HMM) level. For this purpose, the state-level pronunciation modeling concept will first be presented, followed by the corresponding extension to implement a dynamic approach.
- Chapter 6 will introduce the concept of symbolic speaker adaptation applied to the problem of modeling multiple dialects and foreign accents.
- Chapter 7 will describe the basic experiments related to symbolic speaker adaptation. These experiments will then be extended to include additional features and more thorough tests will be made.
- Chapter 8 will conclude this dissertation by a global summary, the list of contributions and some directions for future work.

Chapter 2

Basics of automatic speech recognition (ASR)

In this chapter, we will review the basics of automatic speech recognition systems necessary to understand the content of this dissertation. This chapter is organized in the following manner:

- Section 2.1 will give a general overview of a speech recognition system.
- Section 2.2 will explain how the most relevant characteristics of an input speech utterance can be extracted for speech recognition.
- Section 2.3 is dedicated to the description of Hidden Markov Models, which are the most popular technique applied to speech recognition.

Some remarks and notations found in this chapter are inspired from several references ([114], [145], [158], [2], [55]).

2.1 General overview

In a conversation between two humans, three phases are needed to correctly manage the flow of the dialogue. First, given an utterance of the speaker, the listener must be able to *identify* the correct sequence of words that he/she has just heard. Second, he/she must be able to *understand* the meaning of the sequence. Third, he/she must *speak* in turn to respond to the first speaker in a way that fits well with the dialogue context. Similarly, the same three phases are needed when we would like to establish a conversation between a human and a machine. In the research field, speech identification, understanding and generation are respectively called automatic speech recognition (ASR), automatic speech understanding (ASU) and automatic speech synthesis (ASS). Although all these phases are required, ASU and ASS are beyond the scope of this dissertation and will therefore not be considered. An overview of automatic speech understanding and a study about how ASR and ASU can be integrated can be found in [113]. Readers interested in automatic speech synthesis can refer for example to [37].

The objective of an ASR system is to get the correct sequence of words given a speech utterance. Although it is an easy task for a human, things are more complex for a machine. Humans have the capacity of integrating several types of knowledge (*e.g.*, topic of the conversation) to reliably identify what a speaker said, even in fairly noisy conditions. A standard

ASR system does not have this capacity and must be able to find the correct sequence only from the input speech. In order to facilitate this task, some intermediate procedures are needed.

Let us first consider the speech utterance. It contains of course all information required to correctly identify the underlying sequence, but also some non-relevant information. For example, the pitch, which is responsible for speech tone and tells for instance whether the speaker is male or female, is generally not necessary for recognition and is often removed[1]. Another reason to preprocess the input speech is because the data needs to be compressed for better management by the ASR system. The speech preprocessing step will be described in detail in section 2.2.

Let us now consider the system's output and let us for the moment simplify the problem by assuming that only a single word must be recognized (*e.g.*, a digit between 0 and 9); recognition of sequences of words will be treated later (section 2.3.7). In order to determine what the best word is, the ASR system must represent each word by a model with some characteristics that are different from the other word models. Several types of models exist, but the most popular system is the *Hidden Markov Model* (HMM). Description of HMMs will be given in section 2.3.

To summarize, the task of an ASR system is to identify the correct word from the input speech by extracting some relevant features from the utterance and by matching them against some models representing the possible words, as shown in Figure 2.1.

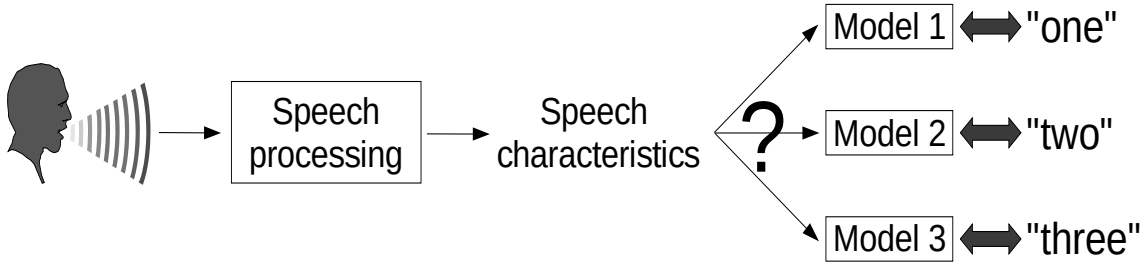

Figure 2.1: General overview of an ASR system

2.2 Extraction of speech characteristics

In order to extract the most relevant characteristics of an input speech, several intermediate steps need to be respected as described below:

Speech acquisition : the speaker talks to the ASR system through a microphone that converts the utterance into an electric signal.

Analog-numeric conversion : since a computer cannot handle continuous values, the analog signal needs to be converted into a digital signal. In the first step, the signal is sampled. However, in order to respect the sampling theorem, which requires that the sampling frequency must be at least twice the highest frequency component present in the signal, the latter must be band limited beforehand by a low-pass filter. The sampling frequency is in practice chosen between 8 and 16 kHz (a sampling at 8 kHz is compatible

[1] The pitch is nevertheless relevant for some specific languages as well as for speaker identification and speech understanding.

with standard telephonic frequency bands). Then, samples are quantized and coded in order to be representable with a limited number of bits (generally between 8 and 16).

Windowing : although it should theoretically be possible to extract the speech characteristics in the time domain, more useful information for recognition can be brought to the fore in the frequency domain, so the Fourier transform is applied to the digitalized signal. However, the shape of the vocal tract constantly changes during speech and hence the corresponding signal cannot be considered stationary (its statistical values such as mean and variance change with time). Since the Fourier transform requires the stationarity of the analyzed signal, the latter is segmented in short intervals during which the hypothesis of quasi-stationarity is assumed (even though it is not true for all sounds such as plosives) and the Fourier transform is applied separately on each of these intervals. In practice, each interval is obtained by multiplying the signal in time by a window. A point to determine is the shape of the window to apply, keeping in mind that a multiplication between two signals in time corresponds to a convolution of their spectra in frequency. A rectangular window is not an appropriate choice because its spectrum is a cardinal sinus with important side lobes that substantially transform the spectrum of the signal to analyze. In practice, a *Hamming window* is generally applied because it allies good frequency resolution and reduced side lobes: $\omega_H(n) = 0.54 - 0.46\cos\left(\frac{2n\pi}{N-1}\right)$, where N is the number of samples in the window and $n = 0, ..., N-1$. This windowing process reduces the effective interval of the analysis and therefore successive windows must partially overlap each other to insure a smooth analysis of the whole signal. Typically, a window of about 30ms shifted by about 10ms is generally chosen. The effective analysis interval per window is called a *frame*.

Feature extraction : each frame needs to be transformed in order to get the most relevant characteristics. From this point, different choices of features were proposed in the literature, but the most commonly applied are the Mel-frequency cepstral coefficients (MFCC)[2]. To obtain them, the Fourier transform is first applied to the windowed samples to obtain the corresponding magnitude coefficients in the frequency domain. Then, a bank of partially overlapping triangular filters that are equally spaced in the Mel scale (so different in the linear scale) is applied: each magnitude coefficient is multiplied by the corresponding filter gain and the results are accumulated to yield a filterbank amplitude in each band. These amplitudes are highly correlated, so further transformation to the cepstral domain is necessary to reduce the cross-correlation effects (and to reduce the number of parameters to take account for acoustic modeling, cf. section 2.3.2): the logarithm is taken to get the log filterbank amplitudes m_j, then the cosine transform is applied to yield the MFCC coefficients c_i: $c_i = \sqrt{\frac{2}{N_c}} \sum_{j=1}^{N_c} m_j \cos\left(\frac{\pi i}{N_c}(j-0.5)\right)$, where N_c is the number of filterbank channels.

Between 12 and 16 coefficients are generally kept to remove the pitch information located at higher coefficient orders. The 0^{th} coefficient is often not included because it is sensitive to the acquisition channel gain. The selected coefficients are also often augmented with the energy of the signal as well as their first and second order time-derivatives, which were shown to be useful. All these coefficients are calculated for each window and are put together in a

[2]The Mel-frequency characterizes how a sound with a certain frequency is perceived by a human auditory system. For example, when a 3000 Hz sound is emitted, it is generally perceived by a human as 1800-2000 Hz approximately. The relationship between real and perceived frequencies has been approximated by the following formula: $Mel(f) = 2595\log_{10}(1 + \frac{f}{700})$. It has empirically been shown that a system incorporating this non-linear behavior of the human ear improves recognition performance.

so-called *acoustic vector*, also called *acoustic observation*. Theses vectors represent the basic speech features used for speech recognition.

2.3 Hidden Markov Models (HMM)

2.3.1 Overview

As mentioned previously, the most popular model used in speech recognition is the Hidden Markov Model (HMM), first introduced by Baker [9] and Jelinek [73]. An HMM is a graph with states and oriented arcs that can in principle be configured in any way, but the most common topology in speech recognition is called a *left-to-right* HMM, that is, states are connected in a linear fashion with arcs oriented from left to right. Besides, each state generally contains a self-loop arc to model duration variability of speech. A left-to-right HMM with five states and self-loops is shown in Figure 2.2. Arcs to skip states may also be added.

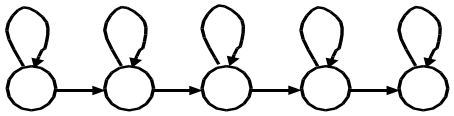

Figure 2.2: Example of a left-to-right HMM

Let us reconsider now the simple word recognition task that was illustrated in Figure 2.1 and let us assume for the moment that each word is associated with a specific HMM (we will see later in section 2.3.3 that it is not always the case). For a given speech utterance, the extracted sequence of acoustic vectors must be matched against the competing HMMs to determine the best model and equivalently the best word. For this purpose, all HMM states and transitions have a certain number of parameters based on statistics that differentiate them from each other. Given these parameters, acoustic vectors are matched against the models by finding an association between the sequence of vectors and the sequences of HMM states. It is said that a state *emits* an acoustic vector when a correspondence is established between a vector and a state. The strength of a vector-to-state association is given by a *probability of emission* given by an observation probability distribution of the state, $b_j(\boldsymbol{o}_t) = p(\boldsymbol{o}_t|q_t = j)$, where $\boldsymbol{o}_t$ is the acoustic observation at time t and q_t is the state variable at time t. Such association represents a time slot corresponding to a frame. In the next time slot, the next acoustic vector is emitted by one of the following states connected to the available transitions (it can be the same state if a self-loop arc is chosen). The transition from one state to another (or the same) is characterized by a *probability of transition* $a_{ij} = P(q_{t+1} = j|q_t = i)$[3]. The best word is given by the HMM with the highest cumulated probability calculated from the probabilities of emissions and transitions resulting from the best associations. An example of vector-to-state associations is shown in Figure 2.3, in which nine acoustic vectors, $\boldsymbol{o}_1$ to $\boldsymbol{o}_9$, have been emitted by five states, s_1 to s_5.

It should be noticed that due to multiple transitions from or to an HMM state, several alternative associations are possible. So even if the best model is known, the best sequence of associations is hidden among the possible choices. This is why Markov Models are called *Hidden* Markov Models. They are also called *acoustic models* since they are directly linked to acoustic observations.

[3] In ASR, a first order Markov model is assumed, that is, the transition probability depends only on the previous state.

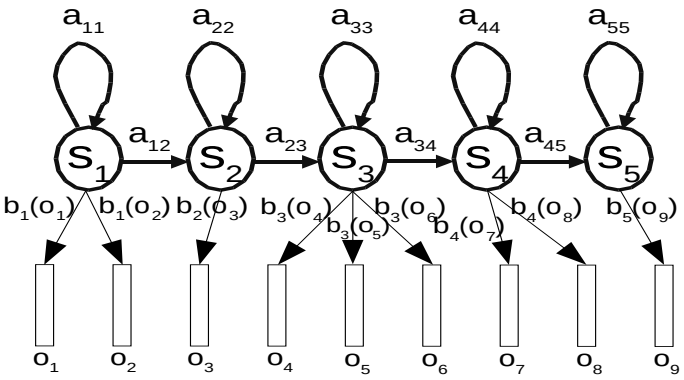

Figure 2.3: Example of associations between acoustic vectors and HMM states

The next section describes how probabilities of emissions and transitions are calculated and how the parameters they depend on are estimated.

2.3.2 Probabilities of emission/transition and training

A point to determine is how to calculate the probabilities of emission and transition. Probabilities of emission can be discrete, semi-continuous or continuous. In the discrete case, a codebook of acoustic vector centroids is defined. All acoustic vectors resulting from an association are replaced by the nearest centroid and the associated probability in the codebook is used as emission probability:

$$b_j(\boldsymbol{o}_t) = P_j\left[\boldsymbol{c}(\boldsymbol{o}_t)\right] \tag{2.1}$$

where $\boldsymbol{c}(\boldsymbol{o}_t)$ is the nearest centroid to the observation (acoustic vector) $\boldsymbol{o}_t$ and P_j is the probability of state j to emit the centroid.

In the continuous case, codebooks are replaced by a multi-dimensional probability density function, typically a Gaussian characterized by its mean and its covariance matrix. It is common to extend the modeling accuracy by using a *Gaussian mixture*, that is, a weighted sum of Gaussians. The probability of an observation $\boldsymbol{o}_t$ to be emitted by the mixture k of a state j at time t is:

$$b_{jk}(\boldsymbol{o}_t) = \frac{1}{(2\pi)^{\frac{n}{2}}\,|\boldsymbol{\Sigma}_{jk}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{o}_t-\boldsymbol{\mu}_{jk})'\boldsymbol{\Sigma}_{jk}^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu}_{jk})} \tag{2.2}$$

where $\boldsymbol{\mu}_{jk}$ and $\boldsymbol{\Sigma}_{jk}$ are the mean and covariance matrix of the mixture k in state j respectively, and n is the dimension of the covariance matrix ($n \times n$). In practice, the covariance matrix is often assumed to be diagonal because MFCCs are not highly correlated, which reduces the number of parameters. The corresponding state-level probability is given by:

$$b_j(\boldsymbol{o}_t) = \sum_{k=1}^{K} c_{jk} b_{jk}(\boldsymbol{o}_t) \tag{2.3}$$

where c_{jk} is a positive weight associated with the mixture k in state j ($\sum_{k=1}^{K} c_{jk} = 1$) and K is the number of mixtures in state j.

In the semi-continuous case, a pool of density functions is shared by all states. Discrimination between states is still possible by using different mixture weights. For a pool of M Gaussians, the emission probability distribution for state j is therefore:

$$b_j(\boldsymbol{o}_t) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{2.4}$$

where c_{jm} is a positive weight associated with the m-th mixture of the pool in state j ($\sum_{m=1}^{M} c_{jm} = 1$), and $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ are the mean and covariance matrix of the m-th mixture of the pool respectively.

Parameters of probability density functions and probabilities of transitions must be estimated based on some training data; the more parameters to estimate and the more training data is required. Training is performed by mapping the extracted acoustic vectors to the sequence of HMMs corresponding to the sequence of correct words in the utterance, then by estimating the HMM parameters from the associations so that the likelihood of the training data given these parameters is maximized. Namely, let us assume a set of unknown parameters to estimate $\theta = (A, B, \pi)$, where $A = \{a_{ij}\}$ is the set of transition probabilities, $B = \{b_j(o_t)\}$ is the set of observation probability distributions and $\pi = \{\pi_i\} = \{P(q_1 = i)\}$ is the initial state distribution. The objective is to estimate the parameters that maximize the likelihood of an observation sequence $\boldsymbol{O}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ p(\boldsymbol{O}|\theta) \tag{2.5}$$

As mentioned previously, the sequence of acoustic observations must be mapped to the sequence of HMM states. We can therefore express the above likelihood as a sum over all possible state sequences $\boldsymbol{q}$:

$$p(\boldsymbol{O}|\theta) = \sum_{all\ \boldsymbol{q}} p(\boldsymbol{O}|\boldsymbol{q}, \theta) P(\boldsymbol{q}|\theta) \tag{2.6}$$

As shown above, two stochastic processes are given by the probabilities in the sum expression. The first probability $p(\boldsymbol{O}|\boldsymbol{q}, \theta)$ is the cumulated probability of emissions that depends on the observation probability distributions $b_j(\boldsymbol{o}_t)$, while $P(\boldsymbol{q}|\theta)$ is the cumulated probability of transitions given as a function of the a_{ij}'s and π_i. Assuming that acoustic observations are independent, we can write:

$$p(\boldsymbol{O}|\boldsymbol{q}, \theta) = b_{q_1}(\boldsymbol{o}_1) \cdot b_{q_2}(\boldsymbol{o}_2) ... b_{q_T}(\boldsymbol{o}_T) \tag{2.7}$$

$$P(\boldsymbol{q}|\theta) = \pi_{q_1} \cdot a_{q_1 q_2} \cdot a_{q_2 q_3} ... a_{q_{T-1} q_T} \tag{2.8}$$

A logarithmic form is used in practice to avoid the too small numbers resulting from successive multiplications. Some popular training algorithms exist, such as the *Baum-Welch* algorithm that iteratively re-estimates the parameters given the training data and the corresponding sequence of word or sub-word labels that refer to the HMMs to train. The algorithm will be described in section 2.3.6.

2.3.3 Phoneme and lexicon

Up to this point, it was assumed that a specific HMM was designed for each distinct word. Although such level of modeling is possible when the number of words to recognize is small (*e.g.*, digit recognition), it becomes impractical when this amount is substantially bigger due to the consequently high number of model parameters to train and estimate. So in practice, each word is decomposed into more elementary units. Several different units have been experimented (*e.g.*, syllables), but the most popular unit is the *phoneme*. A phoneme is the most elementary unit that distinguishes the meaning of a word from another, that is, a change of a phoneme implies a change of the meaning of the word as well. For example, the words "cat" and "hat", respectively transcribed as /k ae t/ and /hh ae t/, are only different by a single phoneme, /k/ vs. /hh/, thus the two words have different meanings. Any word can be transcribed as a sequence of phonemes. The great benefit of this approach is that the number of possible phonemes per language is limited, between 30 and 50 in most cases. Each phoneme can therefore be modeled conveniently by a separate HMM. Although different topologies have been experimented (*e.g.*, [97]), the most popular phoneme-level topology is a left-to-right HMM with three states and self-loop transitions. All words can be modeled by concatenating the appropriate sequence of phoneme HMMs. An example is given in Figure 2.4 for the word "zero", acoustically modeled by concatenating the HMMs of /z/, /iy/, /r/ and /ow/.

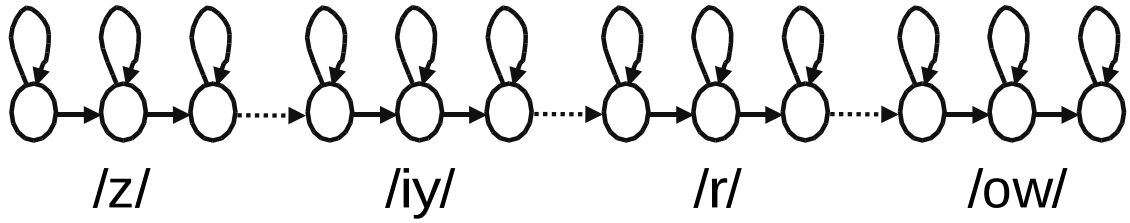

Figure 2.4: Acoustic model of the word "zero" by concatenating phoneme HMMs

When a single HMM is built per phoneme, their representing units are called *monophones*. In practice, even if two words contain the same phoneme, they may be pronounced differently. One of the reasons is because phonemes are highly *coarticulated*, that is, their pronunciations are influenced a lot by their left and right contexts. This is why, in practice, several HMMs are built for a same phoneme to account for the different possible contexts. For example, the word "zero" can be transcribed as /z iy r ow/, but also as /z+iy z-iy+r iy-r+ow r-ow/ ('-' to indicate a left context and '+' for a right context) when the neighbor contexts are explicitly taken into account. Units with a single context (*e.g.*, /z+iy/, /r-ow/) are called *biphones* and those with both left and right contexts included (*e.g.*, z-iy+r, iy-r+ow) are called *triphones*. Performance of an ASR system can be substantially increased if context-dependent models are used, under the condition however that enough data is available to train the increased number of models. Although not all contexts exist in a given language, the number of possible triphones is still substantially high. This is why a clustering procedure is often applied to gather acoustically similar states or models during the training phase.

In this phoneme-level HMM approach, the ASR system needs to know how a word is mapped to a sequence of phoneme models and needs therefore to keep a list of word-to-phoneme correspondences. Such list is called a *lexicon*. A part of a lexicon is shown in Table 2.1[4].

[4] A word-level HMM approach can also use a lexicon in which each word and corresponding transcription are identical.

Word	Transcription
cab	/k ae1 b/
cabinets	/k ae1 b ix n ix t s/
cable	/k ey1 b el/
cafe	/k ae f ey1/
cafeteria	/k ae2 f ax t ih1 r iy ax/
close	/k l ow1 s/
close	/k l ow1 z/
to	/t uw1/
too	/t uw1/
two	/t uw1/

Table 2.1: Example of an ASR lexicon

The "1" and "2" next to some of the phonemes are *stress marks*. In state-of-the-art speech recognition systems, many entries have only one standard transcription (also called *canonical transcription*) per word (*e.g.*, entries from "cab" to "cafeteria"), which is a limitation for speech recognition (we will come back to this problem in the next chapter). ASR lexicons may also include some ambiguities. For example, the two entries for "close" are two *homographs*: the two entries have different meanings (*e.g.*, "close to the post office" (adj.) vs. "to close the door" (verb)) and phonemic transcriptions, but the same orthographic spelling. In practice, some additional marks can be appended to the original spellings to distinguish them from each other (*e.g.*, "close_adj" vs. "close_v"). Another example of ambiguity is the *homophone*, illustrated by the words "to", "too" and "two": they have different meanings and spellings, but the same phonemic transcription. If a specific HMM were built for each word, their acoustic models would be different and homophones could be distinguished from each other, but with a phoneme-level approach their acoustic models are identical, which makes the task of the ASR system more difficult. The problem can partially be solved by the use of a *language model*, which is the subject of the next section.

2.3.4 Language model

Let us now shift from the recognition of a single word to the recognition of a sequence of words. In the speech community, such task is commonly called *continuous speech recognition* (CSR). If nothing is specified, it is assumed that at any time any word may follow another. Although an ASR system with such assumption may work for small vocabulary recognition tasks, (*e.g.*, digit recognition), it becomes impractical for medium and large vocabulary tasks, not only in terms of recognition accuracy, because leaving any word to follow another would yield more sequences that are not well-formed with respect to the given language, but also in terms of recognition speed since all possible word sequences must be tested during recognition (if no explicit pruning is applied). The set of possible successors for a given word must therefore be limited by applying some syntactic constraints. If the number of words is not too big, it is possible to use a *finite-state grammar*, which is basically a graph that represents the possible sequences of words that the ASR system may output. When the number of words is much bigger, it becomes too difficult to design such a network, so a statistical grammar is used instead, where the possible word sequences and associated probabilities are automatically determined from the training material. The component of an ASR system that controls these sequences of words and probabilities is called a *language model*. The probability that a word w_i occurs given $N-1$ previous words is called an *N-gram* and is estimated by counting the

number of times the word w_i follows the $N-1$ previous word sequence over the total number of times the sequence occurs in the training data:

$$\hat{P}(w_i|w_{i-N+1},...,w_{i-1}) = \frac{count(w_{i-N+1},...,w_{i-1},w_i)}{count(w_{i-N+1},...,w_{i-1})} \tag{2.9}$$

The most common language model orders used in ASR are with $N=2$ or $N=3$, respectively called *bigrams* and *trigrams*. In case a sequence of words did not occur often enough to make a reliable probability estimation, a *back-off* procedure is used. Typically for trigrams, when the number of occurrences of the word sequence "w_1, w_2, w_3" is below a certain threshold, the trigram probability $P(w_3|w_1,w_2)$ is replaced (backed-off) by the bigram probability $P(w_3|w_2)$. If even the sequence "w_2, w_3" occurred too infrequently in the training database, then the bigram is backed-off in turn to the unigram probability $P(w_3)$.

The predictability of a sequence of words in a language model is given by its *perplexity*, which is a measure related to the entropy of the grammar. It corresponds approximately to the average number of words that can follow a given word. So the higher the perplexity, the more difficult is the recognition task.

2.3.5 Overall ASR system and recognition

Now that the basic components of an ASR system have been described, let us summarize by putting them together. A standard speech recognition system includes three main components, as shown in Figure 2.5: some acoustic models, a lexicon and a language model. Each component has a specific task. The acoustic models map the input speech utterance to the most likely word or sub-word units via some acoustic feature vectors generated from speech. In case sub-word units are used (*e.g.*, phone), the lexicon governs how they must be combined to form words. The language model tells which sequences of words are allowed to form the output sentence.

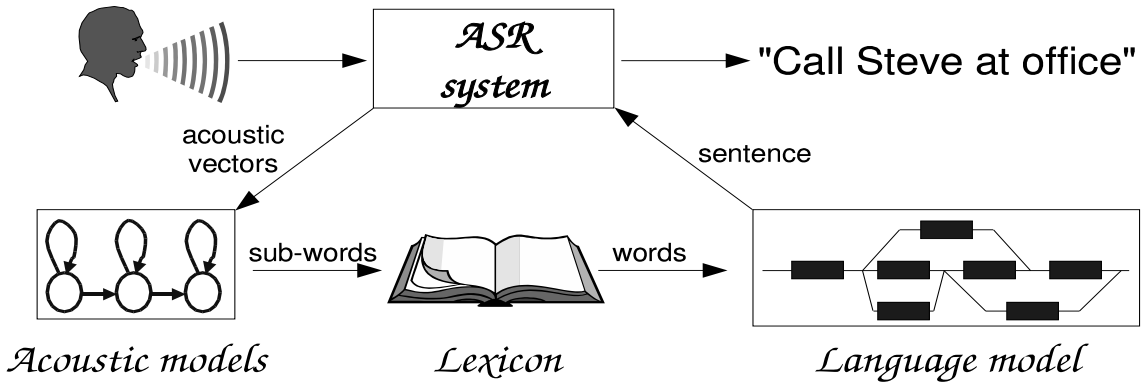

Figure 2.5: Components of an ASR system

Let us see now how these components influence the recognition process. To simplify the problem, let us first consider the isolated word recognition. The basic objective of an ASR system is then to find the correct word given an input speech utterance. In practice, it is implemented by searching for the word $\hat{w}_i$ among all possible words w_i that maximizes the *a posteriori* (MAP) probability given a sequence of acoustic observations $\boldsymbol{O}$ generated from the input speech:

$$\hat{w}_i = \underset{i}{\operatorname{argmax}}\, P(w_i|\boldsymbol{O}) \tag{2.10}$$

This probability is difficult to be estimated directly and is transformed using the *Bayes rule*:

$$\underset{i}{\operatorname{argmax}}\, P(w_i|\boldsymbol{O}) = \underset{i}{\operatorname{argmax}}\, \frac{p(\boldsymbol{O}|w_i)P(w_i)}{p(\boldsymbol{O})} \tag{2.11}$$

where $p(\boldsymbol{O}|w_i)$ is the likelihood of the sequence of observations $\boldsymbol{O}$ given the word w_i and is provided by the acoustic models. $P(w_i)$ is the a priori probability of the word w_i and is provided by the language model. $p(\boldsymbol{O})$ is the probability of the observation and can be omitted since it is the same for all words w_i. Therefore, the objective is to find the word w_i that maximizes $p(\boldsymbol{O}|w_i)P(w_i)$.

Let us see how the acoustic likelihood $p(\boldsymbol{O}|w_i)$ can be calculated. Each word w_i is represented by an acoustic model M_i (which is eventually obtained by concatenation of sub-word models) characterized by its set of parameters $\theta_i = (A_i, B_i, \pi_i)$, where A_i is the set of transition probabilities, B_i is the set of observation probability distributions and π_i is the initial state distribution for the model M_i, respectively. We can therefore establish the following equivalence:

$$p(\boldsymbol{O}|w_i) = p(\boldsymbol{O}|\theta_i) \tag{2.12}$$

We already mentioned that acoustic observations are matched against an HMM by associating the sequence of observations to a sequence of the model states. Since several state sequences are possible, the acoustic likelihood is calculated by summing over all possible sequences $\boldsymbol{q}$:

$$p(\boldsymbol{O}|\theta_i) = \sum_{all\,\boldsymbol{q}} P(p(\boldsymbol{O}|\boldsymbol{q}, \theta_i)P(\boldsymbol{q}, \theta_i) \tag{2.13}$$

which is identical to equation (2.6), but the objective is different since the previous goal was to estimate the acoustic parameters θ_i whereas in the current situation they are supposed to be known and are used to find the best word. We already mentioned that the two stochastic processes found in the sum expression correspond to probabilities of emissions $b_j(\boldsymbol{o}_t)$ and transitions a_{ij}, so the acoustic likelihood can be expressed as follows for an observation sequence $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_T\}$:

$$p(\boldsymbol{O}|\theta_i) = \sum_{all\,\boldsymbol{q}} \pi_{q_1} b_{q_1}(\boldsymbol{o}_1) \cdot a_{q_1 q_2} b_{q_2}(\boldsymbol{o}_2) ... a_{q_{T-1} q_T} b_{q_T}(\boldsymbol{o}_T) \tag{2.14}$$

The above likelihood is difficult to be calculated directly. On the other hand, it can be determined using an iterative procedure described in the next section.

2.3.6 The Baum-Welch algorithm

The iterative procedure to estimate the likelihood in (2.14) is called *forward recursion*. It is actually a part of the *Baum-Welch* algorithm, also called *forward-backward* algorithm. Let us

define the *forward probability* $\alpha_j(t)$ as the joint probability of observing the acoustic vectors $\boldsymbol{o}_1$,...,$\boldsymbol{o}_t$ and of being in state j at time t, given a set of parameters $\theta = (A, B, \pi)$ of a model:

$$\alpha_j(t) = P(\boldsymbol{o}_1, ..., \boldsymbol{o}_t, q_t = j|\theta) \tag{2.15}$$

The values of α for different times t and states j can be obtained through an iterative procedure. For the first acoustic observation, we have:

$$\alpha_i(1) = \pi_i b_i(\boldsymbol{o}_1) \tag{2.16}$$

with $1 \leq i \leq N$ and N is the number of states. The probability of observing the t first acoustic vectors and of emitting the last vector $\boldsymbol{o}_t$ in state j can be obtained by considering all possible paths that could have emitted the previous acoustic vector $\boldsymbol{o}_{t-1}$ in a previous state i, weighted by the transition probability a_{ij} to move from state i to state j:

$$\alpha_j(t) = \left[\sum_{i=1}^{N} \alpha_i(t-1)a_{ij}\right] b_j(\boldsymbol{o}_t) \tag{2.17}$$

with $1 \leq t \leq T$ and $1 \leq j \leq N$. It is required that the last observation is emitted in the last state of the model, hence we have for the likelihood of the whole observation:

$$P(\boldsymbol{O}|\theta) = \alpha_N(T) = \left[\sum_{i=1}^{N} \alpha_i(T-1)a_{ij}\right] b_N(\boldsymbol{o}_T) \tag{2.18}$$

It is therefore possible to evaluate the likelihood of the total observation $\boldsymbol{O}$ given the model parameters θ with the forward recursion.

For the sake of completeness, let us define also the *backward probability* $\beta_j(t)$ as the conditional probability of observing the sequence $\boldsymbol{o}_{t+1}, ..., \boldsymbol{o}_T$ given that we are in state j at time t and given the model parameters θ:

$$\beta_j(t) = P(\boldsymbol{o}_{t+1}, ..., \boldsymbol{o}_T|q_t = j, \theta) \tag{2.19}$$

The values for β can be obtained through the following iterative procedure. The iteration begins with:

$$\beta_i(T) = 1, \quad 1 \leq i \leq N \tag{2.20}$$

The conditional probability of observing $\boldsymbol{o}_{t+1}, ..., \boldsymbol{o}_T$ given that we are in state i at time t is obtained by considering all possible paths that could emit the next acoustic vector $\boldsymbol{o}_{t+1}$ in a following state j, weighted by the transition probability a_{ij} to move from state i to state j:

$$\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{o}_{t+1})\beta_j(t+1) \tag{2.21}$$

As mentioned in section 2.3.2, the Baum-Welch algorithm is the method of choice to iteratively re-estimate parameters of HMMs through a training procedure. This is done with

the use of the forward and backward probabilities α and β. The re-estimation procedure is quite complex and is not the focus of this dissertation, it will therefore not be described here. Interested readers can find more information in *e.g.*, [114]. The re-estimation formulae for means and covariance matrixes are shown below (without proof):

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{t=1}^{T} L_j(t)\boldsymbol{o}_t}{\sum_{t=1}^{T} L_j(t)} \tag{2.22}$$

$$\hat{\boldsymbol{\Sigma}}_j = \frac{\sum_{t=1}^{T} L_j(t)(\boldsymbol{o}_t - \boldsymbol{\mu}_j)(\boldsymbol{o}_t - \boldsymbol{\mu}_j)'}{\sum_{t=1}^{T} L_j(t)} \tag{2.23}$$

where $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ are the re-estimated mean and covariance matrix respectively and $L_j(t)$ is a function of the forward and backward probabilities α and β:

$$L_j(t) = P(q_t = j|\boldsymbol{O}, \theta) = \frac{P(\boldsymbol{O}, q_t = j|\theta)}{P(\boldsymbol{O}|\theta)} = \frac{\alpha_j(t)\beta_j(t)}{P(\boldsymbol{O}|\theta)} \tag{2.24}$$

2.3.7 The Viterbi algorithm

The previous section showed that the likelihood of the observation given the model parameters, $P(\boldsymbol{O}|\theta)$, could be obtained through the forward procedure of the Baum-Welch algorithm. However, this procedure does not generalize easily to the recognition of a sequence of words, because it does not look for a single optimal path, but for all possible paths. In practice, the *Viterbi algorithm* is used instead. The difference compared to the Baum-Welch algorithm is that the sum expression over all possible paths is replaced by the maximum operator. Namely, we are looking for the sequence of states $q_1, ..., q_{t-1}$ that yields the maximum likelihood $\delta_j(t)$ of observing the sequence $\boldsymbol{o}_1, ..., \boldsymbol{o}_t$ and of being in state j at time t:

$$\delta_j(t) = \max_{q_1,...,q_{t-1}} P(\boldsymbol{o}_1, ..., \boldsymbol{o}_t, q_1, ..., q_t = j|\theta) \tag{2.25}$$

The iterative procedure begins with the following initialization:

$$\delta_i(1) = \pi_i b_i(\boldsymbol{o}_1), \quad 1 \leq i \leq N \tag{2.26}$$

With the sum symbol of equation (2.17) replaced by the maximum operator, the partial likelihoods can be obtained through the following recursive expression:

$$\delta_j(t) = \max_i \left[\delta_i(t-1)a_{ij}\right] b_j(\boldsymbol{o}_t) \tag{2.27}$$

It is common to take the logarithm of the above expression because multiplications of probabilities yield too small numbers. The log-likelihood expression is therefore:

$$\log(\delta_j(t)) = \max_i \left[\log(\delta_i(t-1)) + \log(a_{ij})\right] + \log(b_j(\boldsymbol{o}_t)) \tag{2.28}$$

The Viterbi algorithm is implemented by using a trellis. Let us first consider the case of isolated word recognition. An example of trellis is shown in Figure 2.6. Each unit of the horizontal axis corresponds to an acoustic vector representing a time frame, and each unit of the vertical axis represents an HMM state. The objective is to find the best path through the trellis by associating the sequence of acoustic vectors to a sequence of HMM states. The search for the best path is done from left to right, column-by-column, based on emission log-probabilities $\log(b_j(\boldsymbol{o}_t))$ for each vector-state pair, represented by the big dots in the trellis, and on transition log-probabilities $\log(a_{ij})$ associated with the links shown in the figure. At each frame, the partial cumulated log-likelihoods are calculated for all elements of the corresponding column, based on the cumulated log-likelihoods of the previous column and on the transition probabilities connecting the different elements of the previous and current columns. Namely, the cumulated log-likelihood of an element j, $\log(\delta_j(t))$, of the current column is obtained, as shown in equation (2.28), by selecting the element of the previous column with the highest sum of cumulated log-likelihood, $\log(\delta_i(t-1))$, and log-probability of transition, $\log(a_{ij})$, and by adding to it the log-probability of emission, $\log(b_j(\boldsymbol{o}_t))$. Once this is done for all elements in the column, the path is extended by one time frame and another iteration begins with the next column. At the end of all iterations, the final cumulated score gives the global likelihood of seeing the entire observation given the word considered.

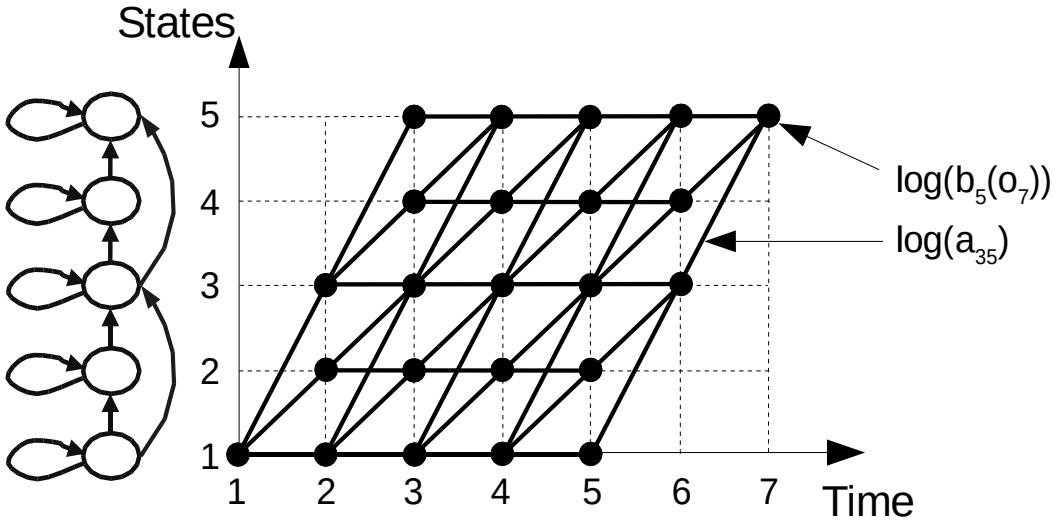

Figure 2.6: Implementation of the Viterbi algorithm using a trellis

In standard recognition, only the best word is required and so only the global likelihood matters, but if the best path through the trellis is also needed, the predecessors selected by the algorithm can also be stored at each iteration so that the best path can be traced back, starting from the upper-right element of the trellis. This process is commonly called *backtracking* and provides *segmentation* information, that is, it tells which acoustic observations are associated with which states. It also gives the best start and end times of the corresponding sub-word units according to the maximum likelihood criterion. A typical application of this procedure is when the sub-word units underlying a speech utterance are known, but not their time boundaries. Determination of the "best" time interval for each unit through this method is called a *Viterbi alignment*, also called *forced alignment*.

Viterbi algorithm for continuous speech

Representation of the Viterbi algorithm using a trellis is also extensible to continuous speech. In this case, acoustic models representing all recognizable words are stacked to form a single column of HMM states (the vertical axis). The sequence of acoustic vectors forms the horizontal axis like with the isolated word case. The trellis can therefore be built using the same procedure as before. However, some additional aspects need to be taken into account:

1. Several words can start a sentence. Therefore, the trellis can also start at different points.

2. Several words can end a sentence. Therefore, the trellis can also end at different points.

3. During the construction of the trellis, the best predecessor of the first HMM state of a word can be itself (through a self-loop transition), but also the last state of any word that can precede the current word. A path can therefore "jump" to transit from one word to another.

The three remarks mentioned above are schematically represented in the left part of Figure 2.7 for three words w_1, w_2 and w_3. It was assumed here that all words can start and end a sentence and any word can follow a given word (including itself). The trellis must therefore be initialized simultaneously at each first state of a possible beginning word and when the last column is reached, the cumulated log-likelihoods of all possible ending words must be compared to determine the best path. Then, the best output sequence of words can be determined through backtracking. The right part of Figure 2.7 shows an example of best path with the sequence "$w_3 - w_1$".

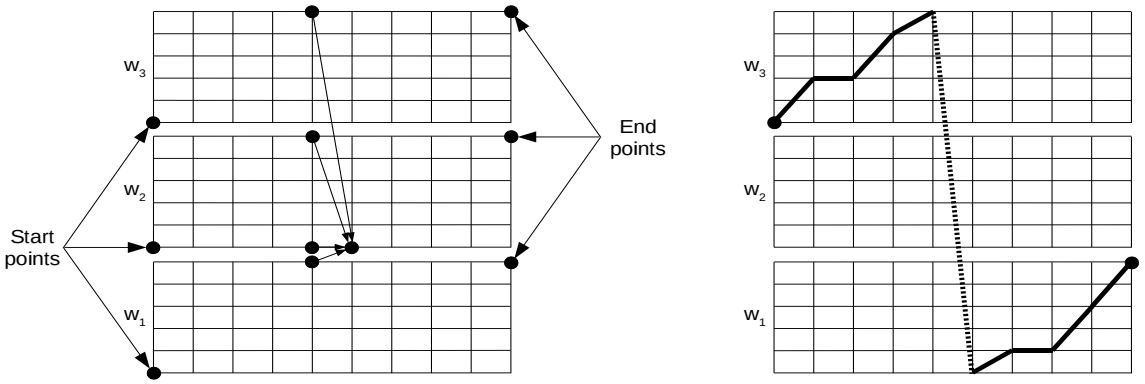

Figure 2.7: Application of the Viterbi algorithm to continuous speech recognition and example of best path

It is sometimes desirable to get not only the best output sequence, but also some alternatives. In this case, the standard Viterbi algorithm must be extended to keep track of more than one predecessor at each word (and eventually state) transition during the decoding process. Since the number of possibilities can be huge, it is common to keep only a subset of *N-best* alternative hypotheses, represented by either a lattice or a ranked list. This dissertation will partly rely on the *token passing algorithm* to generate N-best alternatives; it formulates the Viterbi algorithm as a process of passing tokens through a recognition network. More information about this algorithm can be found in [159].

In practice of course, not all words can start or end a sentence and not all words can follow a given word. This is why the language model (LM) is important in continuous speech, because it limits the number of possibilities by telling which words can follow which other words. The LM can be taken into account in the trellis by adding an LM log-probability to the cumulated log-likelihood each time that a path enters a new word. Besides, two additional factors can influence the overall score. The first factor is the *language model scale*, which modifies the relative importance of the LM relative to acoustic log-likelihoods by multiplying the LM log-probability by this factor. The second factor is the *word penalty*, which is a fixed value subtracted from the overall log-likelihood each time that a path enters a new word; the objective is to avoid insertions of too many words in the output sentence. For instance, if both factors are taken into account and a path went through words w_1 and w_2 and enters a word w_3, the new cumulated log-likelihood $\log(\delta_j(t))$ of state j at time t would be:

$$\log(\delta_j(t)) = \log(\delta_i(t-1)) + \log(a_{ij}) + \log(b_j(\boldsymbol{o}_t)) + S \cdot \log(P(w_3|w_1, w_2)) - P \qquad (2.29)$$

where i is the index of the best previous state, $P(w_3|w_1, w_2)$ is the LM probability for the sequence "$w_1 - w_2 - w_3$" and S and P are the applied LM scale and word penalty respectively.

When only the best sequence of words is required, it is not necessary to keep the list of best predecessors at all iterations. Only transitions from the end of a word to the beginning of the next word are important, namely at which column the best path entered a new word and which was the best predecessor of this word. The best sequence can be retrieved by a recursive process:

1. Beginning from the last element of the best path, get the column where the best path entered the last word.
2. Go to the given column and find which was the best predecessor of the last word.
3. Get the column where the best path entered the best predecessor, and so on.

By this method, word-level segmentation information can also be retrieved at the same place since the columns where the words start are also given.

2.4 Evaluation of ASR systems

In order to measure the performance of an ASR system, we must compare the hypothesized sequences of units (*e.g.*, words) returned by the recognizer with the corresponding reference sequences. This is done by aligning each pair of reference-hypothesized sequences using a dynamic programming algorithm (*e.g.*, [142]) to find the best mappings between their composing units. If a hypothesized sequence does not perfectly match a reference sequence, three types of error can be defined from the alignment made: a *substitution* error when a reference unit is mapped to a different unit, a *deletion* error when a reference unit is not mapped to any hypothesized unit and an *insertion* error when a hypothesized unit is not mapped to any reference unit. These errors are used to compute a metric that evaluates the performance of an ASR system. For word recognition, the two most popular metrics are the *Word Accuracy* (WA) (WA) and the *Word Error Rate* (WER). Given N, S, D and I the number of reference words to recognize, substitutions, deletions and insertions respectively, the WA is defined by:

$$WA = \frac{N - S - D - I}{N} * 100 \tag{2.30}$$

and the WER by:

$$WER = 100 - WA = \frac{S + D + I}{N} * 100 \tag{2.31}$$

The WA and WER can be respectively negative and above 100% because the number of insertion errors is not bounded. Another frequently used metric that does not include insertion errors is the *word correct rate* (WCR):

$$WA = \frac{N - S - D}{N} * 100 \tag{2.32}$$

This dissertation will use the WER as the evaluation metric. Additionally, the *Phone Error Rate* (PER) will also be used, which is calculated in the same way as the WER, but applied to phones instead of words; it will measure how accurately the ASR system targets the true pronunciations of utterances.

When a certain method is applied to modify the characteristics of an ASR system, the performance of the latter can be modified and yield another performance value. However, this difference in performance before vs. after the application of the method is subject to some uncertainty, because the number of samples is finite so that evaluation on another set of samples would generally lead to another performance. To determine if the increase or decrease of ASR performance obtained through the application of a method is *statistically significant*, a confidence interval must be determined around each metric. Size of confidence intervals depends both on the associated metric value and on the size of the test set. The statistical significance test used in this dissertation is described in appendix E.

2.5 Summary

In this chapter, we described the basics of state-of-the-art speech recognition systems. It was first explained how the most relevant speech characteristics for recognition could be extracted from the input speech utterance to form a sequence of acoustic vectors, in particular the Mel-frequency cepstral coefficients, which are the most popular features. Then, Hidden Markov Models (HMM) were introduced as they are the method of choice for speech recognition. The three main components of an HMM-based system were also described: acoustic models, lexicon and language model. These components successively map a sequence of units to another higher-level sequence of units: from acoustic vectors to phones by the acoustic models, from phones to words by the lexicon and from words to sentences by the language model. The two most popular methods for training and decoding with HMMs are the Baum-Welch and Viterbi algorithms.

As it could be expected, standard ASR systems are not without limitations. One of the issues involves the presence of pronunciation variations, especially in spontaneous speech, that state-of-the-art ASR systems do not handle properly. The next chapter will review some of the problems associated with this field and how they were addressed in the literature.

Chapter 3

Basics of pronunciation variation modeling

This chapter will review some basic notions in pronunciation variation modeling for speech recognition. We will describe the limitations of standard ASR systems due to pronunciation variation as well as some methods proposed in the literature to address these issues. The presentation is structured in the following way:

- Section 3.1 will define some terms commonly used in pronunciation modeling (phoneme, allophone, phone).
- Sections 3.2 and 3.3 will explain why pronunciation variation is an important factor to take account and how it can affect ASR systems if it is not properly modeled.
- Sections 3.4 to 3.7 constitute the core part of this chapter and will describe the types of methods proposed in the literature to model pronunciation variation.
- Section 3.8 will explain how the performance of a pronunciation modeling method can be evaluated.
- Section 3.9 will give an insight into some of the current trends in pronunciation modeling.
- Section 3.10 will summarize the chapter.

The content of this chapter is partly inspired from the survey in pronunciation modeling published by Strik and Cucchiarini [135]. Some additions are included and some points will be described in more detail, especially when they are useful to understand the next chapters of this dissertation.

3.1 Phoneme, allophone, phone

Before continuing further, it is useful to define some terms commonly used in the speech community and particularly in pronunciation modeling:

A phoneme (reminder) is the most elementary unit that distinguishes the meaning of a word from another, that is, a change of a phoneme implies a change of the meaning of the

word as well. For example, the words "hat" and "cat", phonemically transcribed as /hh ae t/ and /k ae t/ respectively, are only different by a single phoneme, /k/ vs. /hh/, and the two words have different meanings. A phoneme is an abstract unit that can be pronounced in different ways (see below).

An allophone is a variant of a phoneme that does not change the meaning of a word. A phoneme may be realized as one of several distinct allophones. To better understand the difference between these two terms, a phoneme can be compared to an object (*e.g.*, a table) that can have different colors and shapes (*e.g.*, blue or green, round or square table), but whatever the possible variations, it does not modify the notion of the object itself (*i.e.*, a table is not a chair). For example, the canonical transcription of "hat" is /hh ae t/, but it can be pronounced as [hh eh t] without changing the meaning of the word, hence [eh] is an allophone of /ae/.

A phone is the smallest identifiable speech unit as it is pronounced. This is a more generic term than "allophone", because its meaning in context does not necessarily imply a dependency to a certain phoneme. In other words, the term "phone" can be used to mean either a realization of a phoneme or a simple speech unit (regardless of the phoneme which realized it), while the term "allophone" is more commonly used in the former case.

In the literature, we commonly say that a phoneme may be realized as one or more different phones. When several phones are the realizations of the *same* phoneme, they are called the allophones of this phoneme.

Similar to the distinction between phonemes and phones, a transcription based on phonemes is called a *phonemic transcription* or a *baseform*, conventionally surrounded by "/.../" delimiters, while a transcription based on phones is called a *phonetic transcription* or a *surface form* and is surrounded by "[...]" delimiters.

In the literature however, the terms "phoneme" and "phone" are often used interchangeably because both are frequently used as modeling units of ASR systems. This dissertation will employ both terms, although "phone" will be used as a more generic term than "phoneme"; the latter will generally be used to mark the difference between an abstract unit and its realization.

3.2 Importance of pronunciation variation

State-of-the-art ASR systems perform well when the characteristics and conditions under which the recognition is performed are favorable. On the other hand, recognition performance can drop substantially as soon as the situation gets more difficult. This chapter is dedicated to the analysis of one of its major factors: pronunciation variation.

There are many sources of pronunciation variation. They can be categorized in two types, *inter-speaker* and *intra-speaker* variation. It is indeed intuitive to understand that different speakers do not pronounce words in the same way due to differences in gender, age or social background for instance, but also that a same speaker does not pronounce words in the same manner across utterances either, due to the state of the person (*e.g.*, health, emotional state) and the influence of the environment (*e.g.*, hyper-articulation under noisy conditions). Peters and Stubley [112] projected acoustic trajectories of phone utterances from several speakers on a two dimensional plane and showed that variation can be as high for a single speaker as across speakers.

Early works in ASR were focused on simple tasks like isolated word recognition. But as substantial progress was made and due to the availability of more realistic speech databases to work with, research shifted from the identification of isolated words to the recognition of continuous speech, which comprises carefully read speech, but also more natural (also called spontaneous) speech. The amount of pronunciation variation is substantially higher in the latter case, because people tend to minimize the effort required to transmit a message as long as the listener can get the meaning of it using higher-level knowledge. An analysis made by Fosler-Lussier and Morgan [44] on the development test set of Switchboard (a spontaneous speech database) showed that only 33% of words were canonically pronounced. Their experiments also showed that deviation from canonical pronunciation was greatly influenced by speaking rate and word predictability (frequency). Greenberg [56] phonetically transcribed a portion of the Switchboard corpus and found a lot of possible pronunciations for frequent words, for example more than 80 variants for the single word "and". These analyses show how pronunciation variation has become an important factor to account for. The next section will describe the influence of such variation on ASR systems.

3.3 Limitations of standard ASR systems

3.3.1 Effects of pronunciation variation to ASR systems

At this point, a question to be asked is: what are the possible effects of pronunciation variation to ASR systems ? In fact, these variations are harmful both to the training and recognition phases of a system if they are not modeled properly.

Let us first consider the training phase. As seen in the previous chapter, it consists of iteratively aligning some acoustic models to acoustic observations and of re-estimating the model parameters from the alignments. Choices of models to use are guided by the sequence of correct words found in the utterance. However, if phones are the basic units of the system, it is also desirable to know how these words were *pronounced* in order to use the correct sequence of phone models to align with. The most reliable way is to ask a human expert to phonetically transcribe each speech utterance, but such method requires too much time and expertise. An alternative method is to rely on the ASR lexicon to automatically map the words to a sequence of phone models. However, most state-of-the-art ASR systems contain either a single or only a few alternative pronunciations per word. As a consequence, it is possible that a wrong sequence of acoustic models is used for training. For instance, suppose that an utterance contains the word "had", whose canonical transcription is /hh ae d/, but was actually pronounced [hh eh d]. If the lexicon contains only the canonical transcription /hh ae d/, the latter will nevertheless be used for training. Consequently, the /ae/ model will be contaminated because it will be trained using a wrong ([eh]) speech segment.

Let us consider now another example to understand the potential problem during the recognition phase. Suppose that a speaker uttered the word "command" and pronounced it [k aa m eh n d]. Let us assume that the lexicon only contains the canonical transcription /k ax m ae n d/ for this word, but also the transcription /k aa m eh n t/ for another word "comment". Given these two similar baseforms, the speaker's pronunciation is ambiguous and the ASR system could recognize "comment" whereas the speaker uttered "command". Such ambiguity could be alleviated if, for instance, an additional transcription /k aa m eh n d/ (*i.e.*, that exactly matches the speaker's pronunciation) were associated with the word "command"[1].

[1] As it will be described later in this chapter, addition of such entry does not necessarily mean that the

The effects of pronunciation variation can be detrimental to ASR systems, especially in spontaneous speech. For example, Fosler-Lussier and Morgan [44] reported that an increase of pronunciation variation provoked by a shift in speaking rate yielded a 14% absolute drop in performance on Switchboard. On the other hand, a good match between pronunciations and acoustics can significantly increase the recognition rate. McAllaster et al. [104] simulated some speech data with their acoustic models and found that when all pronunciations of their data exactly matched the transcriptions found in their lexicon, ASR performance increased between 5 and 10 times, depending on the acoustic models they used in their experiments.

3.3.2 Triphones vs. pronunciation variation modeling

Despite the limitations mentioned in the previous subsection, ASR systems are nevertheless able to compensate the effects of pronunciation variation to a certain extent with context-dependent models. Indeed, triphones consider left and right phonetic contexts to account for possible realizations of a phoneme due to coarticulation. The use of multiple Gaussian mixtures per HMM state further increases the modeling capacity.

However, for all that the negative effects of pronunciation variation are not completely eliminated. During training, wrong alignments between models and data will increase the variances of acoustic distributions in HMMs and will consequently require more mixture components to accurately model them. Therefore, complexity of models would be increased unnecessarily. Furthermore, there are some kinds of pronunciation variation that context-dependent models cannot capture very well. Jurafsky et al. [76] showed that while triphones are well-suited for phoneme substitutions and vowel reductions, they do not model accurately complete syllable deletions. Adda-Decker and Lamel [1] found that missing phonemes in transcriptions such as "liaisons" or final schwas in French are particularly prone to errors if they are not properly accounted for in lexicons. The potential effects of triphones with and without pronunciation variation modeling in ASR performance have been brought to the fore by McAllaster et al. [104]. On one hand, they evaluated some triphones trained on Switchboard data and obtained a baseline performance. On the other hand, they generated some simulated data that exactly matched their acoustic models, but using hand-labeled phonetic transcriptions that were significantly different from the canonical transcriptions found in the ASR lexicon. They only obtained a slight increase in performance compared to their baseline. It is only when the pronunciations also matched the transcriptions in the lexicon that the recognition accuracy increased by a factor between 5 and 10 times.

It is therefore important to take account of pronunciation variation in ASR despite the presence of context-dependent models. The next sections will give an insight into some methods reported in the literature.

3.4 Levels and phases of pronunciation modeling

Pronunciation variation can be modeled at all levels of an ASR system: lexicon, acoustic models and language model. The most common approach is however to model it at the lexicon level, since it is convenient to control the variations by modifying or adding new transcriptions. Hereafter, it will be assumed that all methods described are applied to the

overall performance will be improved (risks of lexical confusability), but it would at least avoid a potential error in this particular situation.

lexicon level. Section 3.7 will then give more description about the application of these methods to the other levels (acoustic, language model).

In order to properly model pronunciation variation, two phases are generally required: a *generation* phase and a *selection* phase. The generation phase consists of discovering a set of alternative phonetic transcriptions that represent the possible pronunciations of a word as accurately as possible, while the selection phase consists of keeping only a subset of these transcriptions based on a certain criterion. The two phases will be described in sections 3.5 and 3.6 respectively.

3.5 Generation of pronunciation variants

3.5.1 Knowledge-based vs. data-driven methods

There are basically two ways to obtain pronunciation variants, either knowledge-based or data-driven. *Knowledge-based* methods rely on a priori knowledge on pronunciation variation. Typically, one could refer to some existing pronunciation dictionaries (*e.g.*, [121]) that contain a large set of pronunciation variants per word (although it is not straightforward to use these dictionaries for ASR since they are coded in compact form, see Roach and Arnfield [120] for more details). An alternative practice is to define some general rules based on linguistic studies on pronunciation variation that transform a canonical transcription to obtain the list of possible pronunciation variants or a network of allophones per word (*e.g.*, Wester et al. [149], Cohen [28]). The great benefit of knowledge-based methods is they are general enough to be applicable to any situation. On the other hand, there is always a risk that these techniques do not match well the real pronunciations found in a given speech task and generate either not enough or too many variants. Besides, some knowledge about pronunciation variation of a given language is of course required; such approach becomes difficult when dealing with pronunciation variants of multiple dialects and foreign accents since knowledge of all the corresponding source languages is then needed.

Alternatively, *data-driven* methods can be used, which consist of inferring pronunciation variants directly from speech data. The benefit of data-driven methods compared to knowledge-based approaches is that pronunciation variants will better match the underlying pronunciations of a speech database, but they may lack a certain generality and may not be portable to another ASR system or speech database. Furthermore, when modeling pronunciation variation of multiple dialects and foreign accents, sufficient amount of speech database of each targeted dialect and accent is a priori required, which is difficult to get. Recently however, Goronzy [55] (chapter 8) proposed a new method to automatically derive non-native pronunciation variants using only data from the source and target languages, so without the need of any accented data.

A common starting point of all data-driven methods is to phonetically transcribe the acoustic signal in order to infer the pronunciation variants. This process can be done manually by a phonetician, but it requires a lot of time and expertise and is therefore not feasible for large speech corpora. An alternative method is to retrieve a sequence of phones from the signal using a *phone recognizer*. The great benefit of this approach is that this procedure is automatic and can be applied to a large speech database. On the other hand, it is much prone to errors: while phone recognition accuracy can achieve acceptable performance for some read continuous speech databases (*e.g.*, above 70% with TIMIT), it can easily drop for more complicated tasks (*e.g.*, Humphries [67] obtained only around 50% on the WSJCAM0

database), especially with spontaneous speech databases. This is why some research is focused on improving the quality of phonetic transcriptions (*e.g.*, Cucchiarini and Binnenpoorte [31]).

In the literature, some papers directly use phonetic transcriptions despite their errors, because they can partially be smoothed and pruned through a formalization procedure (described in section 3.5.2) and the selection phase (described in section 3.6). Nevertheless, some other papers try to enhance the simple phone recognition approach. Humphries [67] used confidence measures (*e.g.*, number of competing phone hypotheses over a period of time) to ignore the portions of the recognizer with low confidence scores. The most common approach is however by means of *forced recognition*: it consists of constraining the recognition with a pronunciation network that tells which sequences of phones are allowed to constitute a valid pronunciation variant. This procedure is actually the Viterbi (forced) alignment described in section 2.3.7, although the objective is different: in the previous case, we were looking for the start and end times of each phone given a single sequence of these subword units, whereas in the current situation several distinct phone paths are possible for a given word or sentence and the goal is to find the best subword sequence, often regardless of their time boundaries (although it is still possible to get them as well through this procedure). Since the basic algorithm to find the best path remains the same, this dissertation will use the general term "Viterbi alignment" to refer to both kinds of procedures.

There are several ways to build a pronunciation network. For example, Yang et al. [156] built first a linear network topology containing the canonical transcription of a sentence, but then supplemented it with arcs to account for optional deletions and substitutions of each phoneme and insertions between two successive phonemes. The network also contained transition probabilities and acoustic penalties to control the amount of deviation from the canonical version. Other networks with similar optional modification principle can be found in the literature (*e.g.*, Adda-Decker and Lamel [1], Kessens et al. [79]). Another possibility is to generate the network using knowledge-based phonological rules (*e.g.*, Finke and Waibel [42]); rules should however be designed so that they do not restrict too much the generation of possible variants.

Once a sequence of phones has been obtained from the speech signal, pronunciation variants for each word can be deduced, either directly if each utterance is a single word, or by mapping each word to the appropriate sequence of phones (*e.g.*, by alignment with the canonical transcriptions of words using dynamic programming) if each utterance is a sentence.

3.5.2 Direct vs. indirect pronunciation modeling

Regardless of the type of generation method (knowledge-based and/or data-driven) adopted, pronunciation variation can be represented directly or indirectly. A direct representation consists of simply enumerating the possible pronunciation variants per word. This is typically how they are represented in knowledge-based pronunciation dictionaries or obtained in a data-driven manner from the acoustic signal, as described in the previous subsection. The advantage of enumeration is that it is straightforward and variants can be obtained easily. On the other hand, it is difficult to see how some variations occur across words and hence to control them.

An indirect representation consists of formalizing the possible pronunciation variants, *i.e.*, to build a pronunciation model that represents the variations in a more compact form. A typical example is the construction of pronunciation rules used to transform the canonical transcription of a word to generate a set of pronunciation variants. Such pronunciation models can be designed manually based on relevant linguistic studies, or automatically through a

data-driven method from the available speech database. A common practice in the latter case respects the following points for each training utterance of the database (an example is shown in Figure 3.1):

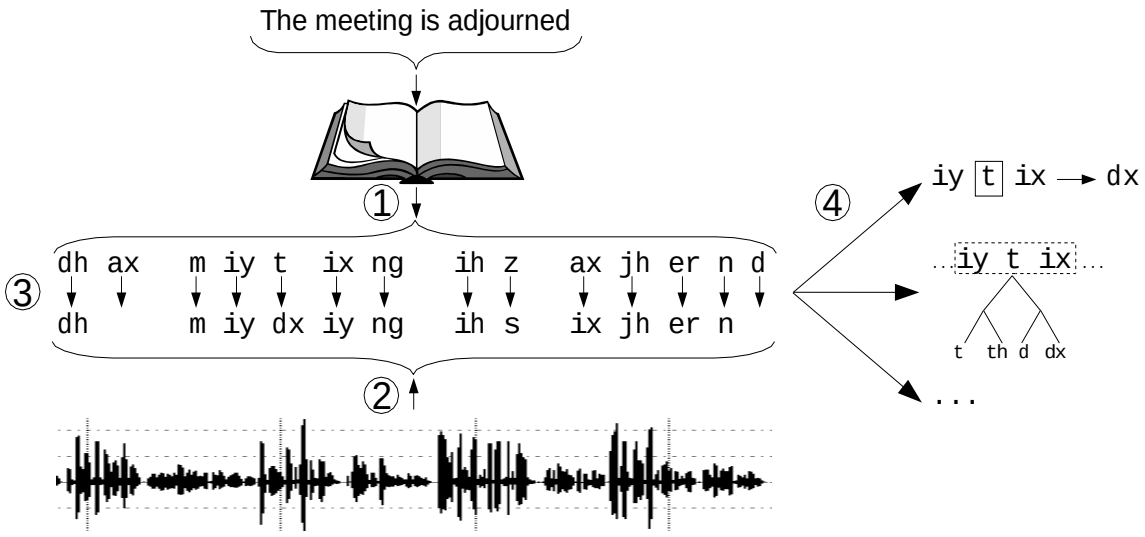

Figure 3.1: Procedure to indirectly model pronunciation variation

1. Given the sequence of words uttered, retrieve the canonical transcription of each word from the ASR lexicon and concatenate them to obtain a sequence of phonemes for the utterance.
2. Phonetically transcribe the acoustic signal (*e.g.*, using a phone recognizer, cf. section 3.5.1) to get a sequence of phones for the same utterance.
3. Align the sequence of phonemes to the sequence of phones to get a list of phoneme-to-phone maps using a dynamic programming algorithm.
4. Use the phoneme-to-phone maps to build a pronunciation model.

Different pronunciation models were proposed in the literature, such as pronunciation rules (*e.g.*, Cremelie and Martens [29]), decision trees (*e.g.*, Riley et al. [118]), neural networks (*e.g.*, Fukada et al. [48]) and confusion matrices (*e.g.*, Torre et al. [139]). The most popular methods (and also used in this dissertation) are pronunciation rules and decision trees, which will be described in the next subsections.

3.5.3 Pronunciation rules

(Note: most notations and terms used in this section are borrowed from Cremelie and Martens [29].)

A general formulation of a pronunciation rule r is given by the following expression:

$$r: \quad L\underline{F}R \rightarrow F' \tag{3.1}$$

This expression means that the *focus* F is realized as the *output* F', but only if F is surrounded by the left context L and the right context R. $L\underline{F}R$ is called the *condition* of the

rule. The focus F must contain at least one phoneme, while the contexts L and R and the output F' can contain zero or more phone(me)s. It is also possible to include a special symbol to represent a word boundary in the rule condition.

A rule is *eligible* if its condition is satisfied, which does not necessarily mean that the rule will be *applied*. Therefore, whenever a rule is eligible, two outputs are generated, one with the rule applied and the other with the rule not applied. To quantify how likely is each output, each rule r is associated with a probability that measures how likely a speaker would realize the focus F as the output F' if the condition of the rule, $L\underline{F}R$, is satisfied:

$$P(r) = P(F'|L\underline{F}R) \tag{3.2}$$

Each rule probability is typically estimated by counting the number of times the rule is applied over the number of times it is eligible:

$$P(r) \cong \frac{Count(L\underline{F}R \rightarrow F')}{Count(L\underline{F}R)} \tag{3.3}$$

These counts can be obtained from the phoneme-to-phone maps obtained in the previous subsection.

Although different methods exist, a simple way of transforming a sequence of phonemes given a set of rules is to verify and apply each rule individually, one after another. Figure 3.2 illustrates an example of process with the canonical transcription of the word "forecast" and three rules (the '#' symbol denotes a word boundary). The leftmost rule in the figure is not eligible and cannot transform the sequence, while the remaining rules are eligible and optionally applied. This is actually the method adopted for the rule-based experiments described in chapter 6; the reader can refer to section 6.4.3 for more detail.

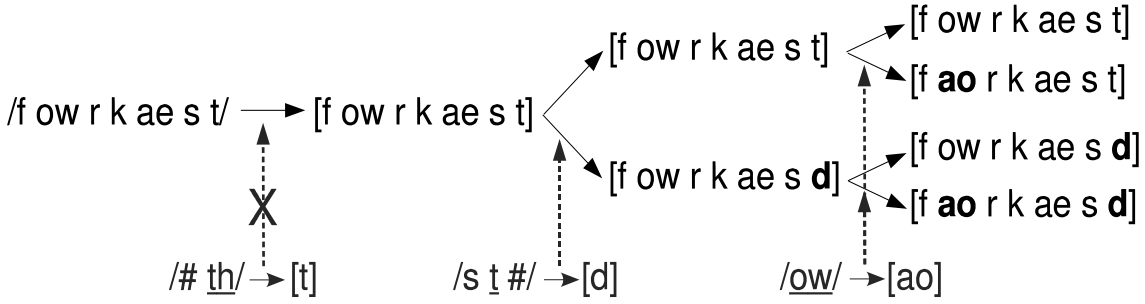

Figure 3.2: Example of generation of pronunciation variants using rules

Besides the rules described above, *negative rules* (also called *exception rules*) can also be defined in order to prevent transformations by the previous rules in some specific contexts. The general form for a negative rule not_r is given by:

$$not\_r: \quad L\underline{F}R \rightarrow not(F') \tag{3.4}$$

which means that when a focus F is surrounded by the left context L and the right context R, it cannot be realized as F'.

3.5.4 Decision trees

A decision tree is a statistical tool that predicts the value of a certain variable given a set of predefined features. In the domain of pronunciation modeling, we are interested in predicting how a certain phoneme in a canonical transcription would be realized under some given contexts. The predicted values are therefore the set of realizable phones given the phoneme. The phoneme and its contexts constitute the set of features provided to the tree. The prediction is carried out by exploring the tree from the root to the leaves and by answering some questions associated with the nodes of the tree and concerning the provided features. Contexts used as features are variable, but the most often used ones are information about the left and right neighbors of the phoneme in the canonical transcription, expressed in terms of their phonetic features. It is also common in practice to use a separate tree per phoneme for a more reliable prediction. Decision trees are well-suited when the number of possible contexts is big, because the classification paradigm into their different branches based on the set of input features lets them also account for contexts not seen in the database used to build the trees.

Training of decision trees consists of providing them with the input features and corresponding predicted value. For pronunciation modeling, they can be obtained from the phoneme-to-phone maps obtained in section 3.5.2. Several algorithms to build decision trees exist in the literature, like CART (Classification And Regression Trees, Breiman et al. [16]) and GRD (abbreviations from the authors, Gelfand, Ravishankar and Delp [53]). The decision trees used in this dissertation will use the CART algorithm, which has been reported to be more effective with limited amount of training data than GRD ([67]). CART consists of building *binary* decision trees, which means that each node of the tree (if not a leaf node) has exactly two children. The two branches connecting a parent to its children answer by a "yes" and a "no" respectively to a question associated with the parent node and concerning one of the input features. More detail about the CART algorithm will be described later in this dissertation during the relevant experiments (section 5.6.1).

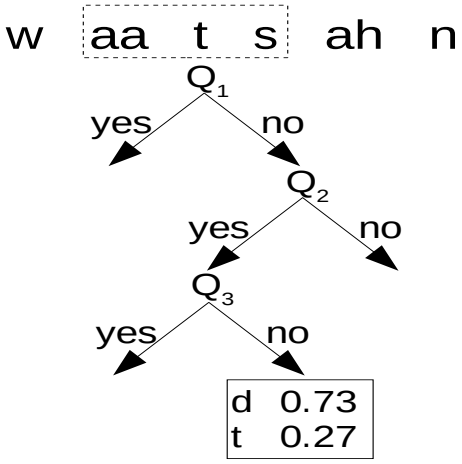

Figure 3.3: Example of generation of pronunciation variants using decision trees

An example of small decision tree with three questions (Q_i, $i = 1, 2, 3$) and using the CART algorithm is illustrated in Figure 3.3. Suppose that we would like to predict the possible realizations of the phoneme /t/ of the proper name "Watson", given the phonetic features of its immediate left and right contexts, /aa/ and /s/ respectively. The decision tree built for the phoneme /t/ will then be explored by asking questions like "Is the right

context phoneme a fricative ?" at each node and by following the branch associated with the corresponding answer. The prediction is given by the leaf found at the end of the exploration, which generally contains a probability distribution of the possible surface form phones, in this example [d] with a probability of 0.73 and [t] with 0.27. Pronunciation variants can be obtained by simply concatenating all possible combinations of successive phones.

3.5.5 Within-word vs. cross-word pronunciation modeling

Pronunciation variation can occur inside a word (*within-word* variation), but also at the junction of two words (*cross-word* variation) in continuous speech. Typically, the "liaison" in French is an example of cross-word variation, since a [z] is added after a word that ends with a [s] or [x] when the next word starts with a vowel[2]. When only within-word variations are modeled, it is sufficient to simply add pronunciation variants to the lexicon. On the other hand, more elaborate method is required to take account of cross-word variations as well, since neighbor words of lexical entries are a priori unknown.

A simple method to take account of cross-word variations is through the use of *multi-words*: it simply consists of adding pairs or even triplets of words as separate entries to the lexicon and of generating their pronunciation variants like with any other word. This method has been successful in some cases (*e.g.*, Finke and Waibel [42]), but not always (*e.g.*, Riley et al. [118], Wester et al. [149]). An alternative approach is to integrate cross-word variations into the recognition network of HMM systems. For instance, Cremelie and Martens [29] built a pronunciation network per word with conditional entries and exits labeled with the index of the rule applied. During the generation of the recognition network - guided by the allowed sequences of words found in the language model - only pronunciation networks of words with compatible entries and exits (*i.e.*, with the same rule index) could be connected to each other. Saraçlar et al. [124] applied a decision tree-based pronunciation model to their phoneme-level recognition network (obtained from the combination of a lexicon and a language model, or from a first recognition pass) to yield a network of surface forms.

3.6 Selection of pronunciation variants

3.6.1 Lexical confusability

A well-known problem in pronunciation modeling is that it is not enough to just generate all possible pronunciation variants and to add them to the lexicon, because *lexical confusability* increases due to an augmentation of similar pronunciation variants across different words, which often leads to a drop in recognition performance. An example is shown in Figure 3.4 for the words "command" and "comment": whereas their canonical transcriptions, /k ax m ae n d/ and /k aa m eh n t/, are distinct by three phonemes, their pronunciation variants, [k ax m eh n d] and [k ax m eh n t], are much closer to each other and add the risk of recognizing "command" as "comment" or vice-versa.

The language and acoustic models can help to avoid some errors to a certain extent, but it is often not enough. It is therefore necessary to reduce the number of possible pronunciation variants or transformations (*e.g.*, number of rules) in order to avoid too much confusability. This is a priori not an easy task, because a variant or transformation that causes many

[2]This is a general rule that contains exceptions, see Adda-Decker and Lamel [1]).

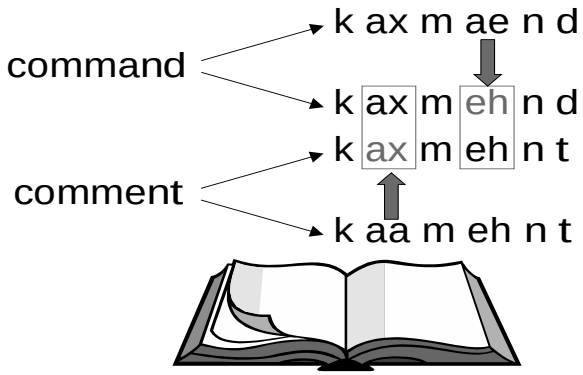

Figure 3.4: Example of lexical confusability between the words "command" and "comment"

recognition errors in some cases can also contribute to considerable improvements in some other cases (Kessens et al. [79]). Some criteria of selection proposed in the literature are listed in the next subsection.

3.6.2 Selection criteria

Maximum frequency : an intuitive choice is to keep only the most frequent variants (*e.g.*, Mokbel and Jouvet [105]) or the most applied transformations (*e.g.*, rules in Wester et al. [149]), based on either a priori knowledge or from occurrences observed in the training data.

Maximum likelihood : since parameters of acoustic models are estimated so that they maximize the likelihood (ML) of the training data, some researchers prefer to select variants or transformations based on the same criterion. Holter and Svendsen [65] and Mokbel and Jouvet [105] grouped tokens of words into clusters based on the ML criterion and represented each cluster by the most likely phonetic transcription (the two papers differ in the clustering algorithm). Amdal [2] selected rules that most increased the total likelihood of word tokens. Variation of likelihood given a word token was measured by the ratio of the log-likelihood of the token *after* applying a rule to the word baseform over the log-likelihood *before* its application.

Confusability measures : some papers proposed methods to eliminate words or phonetic transcriptions that were too confusable. Torre et al. [139] first built a confusion matrix using their unrestricted phone recognizer, then estimated a confusion probability between two transcriptions and between two words based on the matrix. These estimations were used both to select the least confusable words and to generate alternative transcriptions that were as distinct as possible across words. Sloboda [129] also used a confusion matrix to reject pronunciation variants that differed only in highly confusable phonemes. The remaining variants were used to retrain acoustic models with more accurate transcriptions. Roe and Riley [122] measured confusability between pairs of words by taking the intersection of their respective pronunciation networks and by measuring their confusions using the Bhattacharyya distance. Good correlation was obtained between the predictions made by the system and the actual confusions observed. Wester and Fosler-Lussier [147] built a lattice of words that competed each other when their pronunciations matched a same substring of the reference transcription of the data.

Based on the number of competing words in the lattice, they defined two metric bounds that respectively over- and underestimated the number of possible confusions. Recently, Fosler-Lussier et al. [43] extended this work by also taking account of acoustic model confusions and language model and by integrating them into a sequence of weighted finite state transducers (WFSTs, cf. section 3.9).

Confidence measures : alternatively, confidence scores can be used to measure the quality of phonetic transcriptions. Williams and Renals [151] evaluated several metrics and found that frame normalized a posteriori phone probabilities gave the best estimation. They added new pronunciations when their confidence scores were higher than the score of the corresponding canonical transcription of the word.

Entropy : this measure can estimate the uncertainty of a pronunciation given a word. Tsai et al. [140] used it to limit the number of pronunciations per word proportionally to the associated entropy. Following a similar line of thought, entropy can also be related to the uncertainty of applying a transformation to model a pronunciation variation. For instance, Yang and Martens [155] organized their set of pronunciation rules in a hierarchy and iteratively pruned a child rule when its replacement by a parent rule (similar to the child rule, but with shorter rule context) implied only a small change of this uncertainty (measured by the difference of entropy before vs. after pruning)[3].

3.6.3 Dynamic pronunciation modeling

Although reduction in the number of variants or transformations limits the risks of lexical confusability, it also limits pronunciation coverage. This may be a handicap when the amount of pronunciation variation is considerably high, typically when dealing with dialects and foreign accents. An alternative solution in such situation is to let more variants or transformations coexist, but to modify their relative importance or to activate them at different times *during recognition* depending on one or more factors. This is the basic concept of *dynamic pronunciation modeling*, which constitutes one of the major topics of this dissertation.

Examples of dynamic techniques are not frequent, but still exist in the literature. For instance, Fosler-Lussier [45] adopted an N-best list rescoring paradigm: each recognized hypothetic sequence of the list was expanded to a pronunciation network using syllable- and word-level decision trees and based on various measurements (*e.g.*, phonetic features, but also syllable and word-level features, word predictability, speaking rate). Then, a Viterbi alignment on each network defined new acoustic scores, which were used in combination with language model scores to re-rank the hypotheses. Similar idea was applied to lattice rescoring as well, by dynamically determining the pronunciations of words during decoding and by assigning new acoustic scores using decision trees and based on the neighbor words in the lattice and other features. Slight improvement over the static approach was observed. Another example is given by Ostendorf et al. [109] that dynamically selected pronunciation variants or modified their associated probabilities by estimating a "hidden speaking mode" for each utterance, using cues such as speaking rate, normalized energy, etc. They were also intended for rescoring n-best lists or lattices. Finke and Waibel [42] incorporated these cues as questions into decision trees and obtained significant performance improvement on Switchboard. Recently, Ward et al. [144] presented a preliminary work towards dynamic pronunciation modeling. Their experiments suggested that more than two pronunciations per word are necessary for

[3] In a more recent paper [156], they further weighted this metric by an expression containing the application frequency of the child rule in order to prune rules that were not often eligible.

a dynamic approach to be effective, and that prosodic factors like stress and pitch-accent are important cues to model heavy-accented speech. The methods presented in this dissertation also fit in with this framework. However, they are different from the above techniques because they are focused on the extraction of articulatory positions from speech to predict the possible pronunciation variations. Furthermore, we will not only investigate the dynamic approach at the lexicon level (chapter 4), but also at the acoustic (HMM) level (chapter 5).

One of the reasons why dynamic approaches have not often been proposed in the literature is because it unavoidably increases the computation time, since extraction of dynamic cues (*e.g.*, speaking rate) and reduction of pronunciation variants or transformations must be done during recognition. To address this issue to a certain extent, chapter 6 will present an alternative method that builds speaker-dependent lexicons based on a symbolic speaker adaptation technique. Although the technique cannot be considered fully dynamic, it nevertheless allows an adaptation of pronunciation variants to a change of speaker. It will also address the issue of handling multiple dialects and foreign accents.

3.7 Pronunciation modeling at the acoustic and language model levels

Once a set of pronunciation variants or transformations has been obtained, they can be used to augment the lexicon with additional pronunciation transcriptions per word. Additionally, they can update the acoustic models and the language model so that they also account for the possible pronunciation changes.

To update the acoustic models, a common practice is to phonetically retranscribe the training data with Viterbi alignment (forced recognition) that selects among the new added and competing pronunciation variants. These new labels hopefully better match the pronunciations of spoken utterances and are used to retrain the acoustic models (*e.g.*, Sloboda [129]). Iteratively, these new acoustic models can be used to update the pronunciation variants or set of transformations, which in turn can generate another set of phonetic transcriptions, and so on (*e.g.*, Wester et al. [149]). Strik et al. [135] experimentally found however that successive iterations were of limited benefit and a single iteration was enough.

Retraining the acoustic models is not the only way to model pronunciation variation at the acoustic level. Other papers proposed new HMM topologies, new types of units or more explicit acoustic-level pronunciation modeling. Some examples of these techniques will be described later in section 5.2.2 (as they better fit in with a method proposed in chapter 5).

In word recognition, language models (LM) are expressed at the word level, so without any further specification, all pronunciation variants of a word following another word are considered equiprobable. A possible solution to take account of the relative importance between variants is to incorporate the latter into the LM. Wester et al. [149] did this by counting the number of occurrences of each variant in the training data. These counts were then used to build variant-level LMs. Alternatively, probabilities can be associated with pronunciations (while keeping the word-level language model), so that during decoding, a separate pronunciation score can be combined with acoustic and language model scores to influence recognition decisions. Although early experiments reported in this dissertation did not address the LM issue, pronunciation probabilities were then taken into account in later experiments (chapters 6 and 7). In line with Yang et al. [156] who reported that simply adding pronunciation probabilities were not very useful, the latter were also scaled during recognition to emphasize their importance relative

to acoustic and LM scores.

3.8 Evaluation of pronunciation modeling methods

The measure most often used to evaluate the performance of a pronunciation modeling method is the change of word error rate (WER). Namely, given the WER of a baseline ASR system, say WER_{base}, and the new WER of the same system after application of the method, say WER_{new}, performance is evaluated by measuring the difference between the two WERs. This difference can be expressed in absolute terms (ΔWER_{abs}) or in relative terms (ΔWER_{rel}):

$$\Delta WER_{abs} = WER_{base} - WER_{new} \tag{3.5}$$

$$\Delta WER_{rel} = \frac{WER_{base} - WER_{new}}{WER_{base}} \tag{3.6}$$

Although both values are commonly reported in the literature, ΔWER_{rel} is generally preferred because it measures the amount of improvement with respect to the baseline: given a reduction in WER measured with the application of a method, it is more difficult to get this improvement when starting from a high baseline (*i.e.*, with low WER_{base}) than from a low baseline (*i.e.*, with high WER_{base}). ΔWER_{rel} will reflect this difficulty, but not ΔWER_{abs}. Besides this global evaluation method, some researchers (*e.g.*, Kessens et al. [79]) also carry out error analyses to get a better insight into the processes underlying pronunciation variation that improve or deteriorate the performance of their baseline system, but such analyses are less common in the literature.

Although a lot of research has been dedicated to improve ASR performance by modeling pronunciation variations, results reported in the literature so far have been of limited success: around 10% relative reduction in WER and even less most of the time. These results suggest that the right solutions have not been found yet. But as already mentioned in section 3.3.2, several studies (*e.g.*, Adda-Decker and Lamel [1], Jurafsky et al. [76], McAllaster et al. [104]) clearly showed that not only pronunciation modeling is necessary to model certain pronunciation characteristics, but it has also the potential of bringing substantial improvements in spontaneous speech. Besides, some researchers did obtain occasionally larger improvements (*e.g.*, 35% relative reduction of WER in Yang et al. [156]), which encourages more research to target the optimal solutions in the future.

3.9 Some current trends in pronunciation modeling

Although certainly not exhaustive, some noticed trends in pronunciation modeling are listed below:

Dialects and foreign accents : in the past, pronunciation variation was often modeled without considering the speech background of speakers. It is however known that pronunciation variants due to dialects and foreign accents can be detrimental to ASR systems. To the author's knowledge, one of the first works on pronunciation modeling of

dialects and foreign accents[4] was reported by Humphries [67] in 1997. From then, several other papers on this topic were also published (*e.g.*, Amdal et al. [3], Goronzy [55], Huang et al. [66], Kat and Fung [77], Ward et al. [144]). Some examples of works found in the literature on non-native speech will be presented in chapter 6 (section 6.3).

Multilingual ASR : a natural extension of pronunciation modeling of foreign accents is its incorporation into multilingual ASR, which has recently become a popular topic. Korkmazskiy [91] proposed a method to automatically learn letter-to-phoneme mapping rules of any context sensitive language from its pronunciation dictionary or a phonetically transcribed database. Tian et al. [138] compensated the errors of language identification of their multilingual ASR system by getting the N-best identified languages and by applying the corresponding letter-to-phoneme conversion scheme designed for each language to get N pronunciation variants. A pruned set of these variants was then used for speech recognition. Results were comparable to those obtained with a cheating system in which an expert identified the language manually.

Hierarchical structures : they are naturally present in decision trees, which are extensively used for pronunciation modeling. Other types of hierarchies were also proposed in the literature, for example rules (*e.g.*, Cremelie and Martens [29], Korkmazskiy [91]) as well as words and their composing units (*e.g.*, Koval et al. [92], Seneff and Wang [126]), with the objective of modeling pronunciations at different representation levels and/or of pruning some less useful components of the hierarchy.

Weighted Finite State Transducers (WFSTs) : a WFST is an automaton of finite size that maps pairs of strings, possibly expressed with different alphabets, to weights. For example, words and phones can be mapped to a conditional probability distribution of a phone given a word (and its possible pronunciations). Although WFSTs are not new, they were used in several papers of the last workshop in pronunciation modeling (ITRW-PMLA 2002), cf. Caseiro et al. [21], Fosler-Lussier et al. [43], Hazen et al. [63] and Seneff and Wang [126]. One of the tutorials given during the ICSLP 2002 conference also concerned the use of WFSTs in speech recognition. Their properties (*e.g.*, composition of transducers) make them more suitable to handle some phenomena not easily handled by standard ASR systems (*e.g.*, context-dependent modeling at word boundaries). More information about WFSTs can be found for instance in Pereira and Riley [111].

Articulatory information : since it was not proposed in any paper (except in Koval et al. [92]) during the last workshop in pronunciation modeling (ITRW-PMLA 2002), it was considered as a lacking point. Nevertheless, incorporation of articulatory knowledge into ASR has gained much interest, as proved by a special session dedicated to it during the Eurospeech 2001 conference. Although more time is required to acquire better knowledge in this field, there is a good hope that incorporation of articulatory information will greatly help ASR systems in the future. This dissertation follows this line of interest and constitutes a small contribution towards this goal (cf. chapters 4 and 5). Section 4.3 in the next chapter will dedicate a small survey of the literature concerning this topic.

Speaker-dependent pronunciation modeling : since pronunciation variants introduced in a speaker-independent system generally led to only marginal improvements, one current trend is to model pronunciation at the speaker level instead (*e.g.*, Lee et al. [98], Willett et al. [150]). With this approach, higher improvements can be expected, especially if pronunciations are highly variable across speakers. Chapters 6 and 7 of this

[4] *Acoustic* modeling of dialects and foreign accents can be found in earlier works, *e.g.*, in [10].

dissertation will be dedicated to the creation of speaker-dependent lexicons in order to model speakers with different dialects and foreign accents.

3.10 Summary

This chapter presented the basic concepts of pronunciation modeling. This topic has become important, especially since research in ASR started to focus on spontaneous speech that contains a lot more pronunciation variations than carefully read speech. Pronunciation modeling is necessary to both build more accurate models during training and avoid a significant drop in recognition performance. It is also better suited to handle some phenomena (*e.g.*, syllable deletions) that are not well modeled by triphones. Pronunciation variation modeling can be applied to all the basic ASR components (lexicon, acoustic models, language model), although application to the lexicon is the most frequent. Two phases are generally required, a generation phase that consists of discovering new pronunciation variants or transformations to apply, and a selection phase that consists of reducing this initial set in order to limit lexical confusability. Some categories and characteristics of each phase were reviewed. The metric generally used to measure the performance of a pronunciation modeling method is the absolute or relative change of word error rate observed when the method is applied to a baseline system. Finally, some current trends in pronunciation modeling were also described.

The next chapters will describe the methods adopted in this dissertation and experiments carried out to model pronunciation variation. The motivations relative to each method will also be explained in the corresponding chapters.

Chapter 4

Dynamic Lexicon Using Phonetic Features

This chapter will describe a method that dynamically selects a set of pronunciation variants in the lexicon based on the detection of phonetic features [100]. The methodology and some related experiments and results will be presented through the following sections:

- Section 4.1 will expose the key ideas and motivations of the method.
- Section 4.2 will define the notion of phonetic features illustrated by some examples.
- Section 4.3 will present some examples of applications of phonetic features through a literature survey.
- Sections 4.4, 4.5 and 4.6 will describe the methodology in detail.
- Section 4.7 will empirically evaluate the method and report the basic results obtained.
- Section 4.8 will analyze some intermediate results and errors made and will point out the components of the system that could be improved.
- Section 4.9 will expose some ideas on how the issues mentioned in 4.8 could be addressed.
- Section 4.10 will give a summary of the chapter.

4.1 General overview

Two objectives motivated the elaboration of the method described in the next sections:

1. System with both high pronunciation coverage and small lexical confusion.
2. Introduction of articulatory knowledge to the ASR system.

For the first objective, a *dynamic* concept was adopted. It consists of adapting during recognition the pronunciation model to the pronunciation characteristics of a given speech. Considered at the lexicon level, it consists of selecting the lexical entries that best match the way a sentence has been phonetically uttered and of discarding the remaining entries. This approach can not only consider as many pronunciation variants as needed to insure a high

pronunciation coverage, but can also avoid too much lexical confusability by dynamically reducing its content during recognition. The example in Figure 4.1 compares a standard (static) lexicon to dynamic lexicons. Let us assume that each lexicon originally contains two words, "command" and "comment", with two phonetic transcriptions for each of them as shown in the figure. It is clear that these words are highly confusable each other, especially because their pronunciation variants ("command"→[k ax m eh n d] vs. "comment"→[k ax m eh n t]) differ by only a single phone. A standard way to reduce this confusability is to keep only one of these transcriptions, let us say [k ax m eh n d], for *all* sentences. However, this approach could penalize the system's performance with speakers who tend to pronounce "comment" as [k ax m eh n t]. The dynamic approach is different in that both variants are originally kept, but may be activated or discarded at different times during recognition, depending on the input utterance. For instance, assuming that "command" was uttered in a sentence A, [k ax m eh n d] may remain active and [k ax m eh n t] discarded, while the inverse case may happen with a sentence B if "comment" was uttered instead.

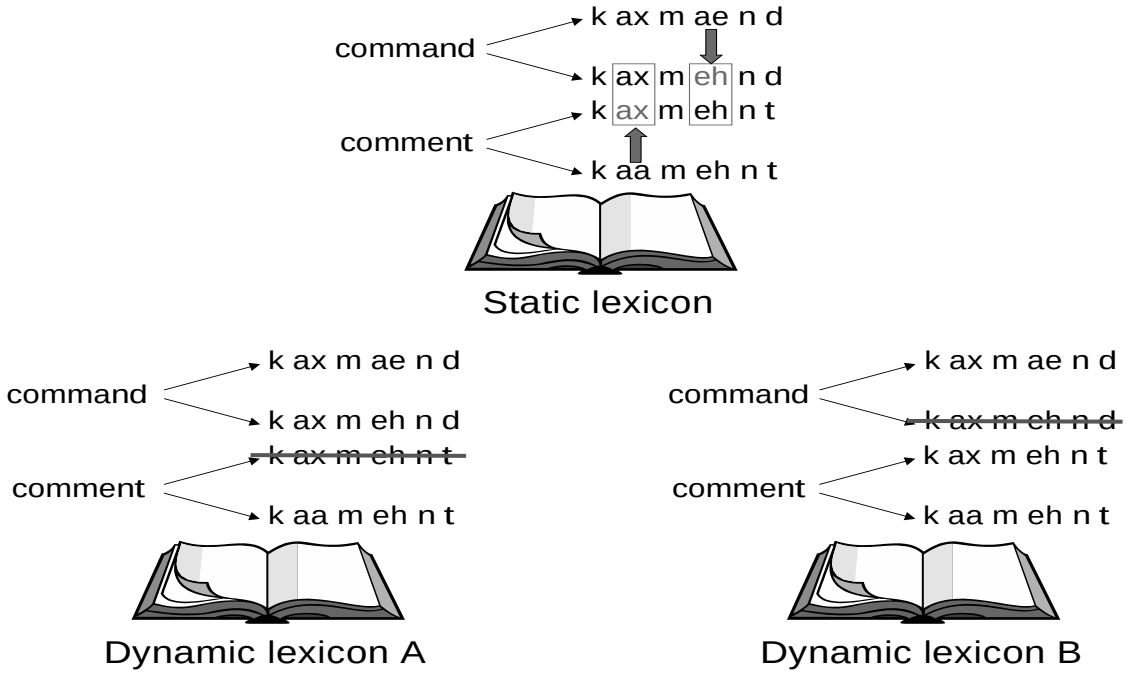

Figure 4.1: Comparative example between static and dynamic lexicons

The second objective (introduction of articulatory knowledge) was originally raised because state-of-the-art speech recognizers (mainly HMMs) are solely based on statistics and do not include any linguistic knowledge at any point. Especially because pronunciation variation modeling was concerned, it seemed desirable to take account of the production mechanism responsible for these variations. For this purpose, our pronunciation modeling approach was based on the detection of *phonetic features*: they are descriptive parameters that differentiate a phonetic unit from another. Phonetic features are of three types: *articulatory features*, which characterize how a speech segment is produced by the human articulatory system (*e.g.*, tongue, lips), *auditory features*, which characterize the physical properties of a speech sound (*e.g.*, grave, acute), and *perceptual features*, which characterize how a sound is perceived by the human auditory system and the brain. Our approach will mainly rely on articulatory features, which are the most well-known and applied in the literature. However, the general term "phonetic features" will generally be employed as most but not all features used in this thesis are based on articulations; the term "articulatory" will nevertheless be used with other

expressions (*e.g.*, articulatory approach, articulatory level,...).

These features helped to decide which lexical entries were retained or rejected to build the final lexicon for a given utterance. The next sections will give a deeper insight into the concept of phonetic features, including their potential benefits and some examples of their utilization.

4.2 Definitions and examples of phonetic features

Phonetic (articulatory) features describe how a speech segment is produced by the human articulatory system. A typical example (commonly called "multi-valued" or "IPA-like" features; IPA stands for "International Phonetic Alphabet" [70]) is given in Table 4.1.

Feature class	Values
Voicing	voiced, voiceless
Place	bilabial, labio-dental, dental, alveolar, postalveolar, retroflex, palatal, velar, glottal
Manner	stop, fricative, approximant, nasal, lateral
Height	high, mid, low
Front-back	front, center, back
Rounding	rounded, unrounded

Table 4.1: Example of phonetic features

In this example, features are grouped in different feature classes, each class representing a different articulatory dimension. Feature classes and their values are chosen so that they are generally independent of each other. The following feature classes have been defined in this table:

- **Voicing** describes the state of the glottis and tells whether the vocal chords vibrate or not during a sound production. Some examples of voiced sounds include [b], [g], [z]. This distinction is usually made for consonants only because vowels are always voiced (except when they are whispered).

- **Place** of articulation tells *which* articulators of the vocal tract are activated when producing a consonant. Such sound is emitted by approaching an articulator to another one, thus making a constriction (narrowing) and cutting the airflow to a certain extent. Two articulators are defined: an *active* articulator that usually moves to make the constriction and a *passive* articulator that usually does not move. For instance, "alveolar" means that the tongue tip or the tongue blade (usually the tip) approaches the alveolar ridge, producing sounds like [d], [n] or [t].

- **Manner** of articulation describes *how* the articulators are involved when producing a consonant sound. "Stop", "fricative" and "approximant" describe the degree of constriction, that is, how close the active articulator gets close to the passive articulator to make the constriction. For example, "fricative" means that the active articulator does not touch the passive articulator, but is close enough to make the air flowing out through the constriction turbulent (*e.g.*, [f], [s], [z]). "Nasal" tells if the soft palate is lowered to let the air to flow out through the nose (*e.g.*, [m], [n]). "Lateral" tells whether the side(s) of the tongue are lowered so that the air can flow out along the side even though active and passive articulators touch each other (*e.g.*, [l]).

- **Height** describes the relative position of the tongue body on the "vertical" axis when producing a vowel sound. Examples of high vowels include [i] and [u]; low vowels include [a] and [ae]. For classification purposes, it is not uncommon to see this category merged with place of articulation in the literature, in which case the term "place" refers to both consonants and vowels.
- **Front-back** describes the relative position of the tongue body on the "horizontal" axis when producing a vowel sound. Examples of front vowels include [ae] and [i]; back vowels include [o] and [u].
- **Rounding** tells whether the lips are rounded during a vowel sound. Examples of rounded vowels include [o] and [u].

Any phone can be expressed by a set of these feature values. For example, the phone [b] is realized when the vocal chords vibrate (voiced), the articulators are the lips[1] (bilabial) that touch each other so that no air flows out (stop), so the phone [b] can be called a "voiced bilabial stop". Not all feature classes are relevant for a given phone. For the example given in Table 4.1, *voicing*, *place* and *manner* are only relevant to consonants, while *height*, *front-back* and *rounding* only refer to vowels. Phones and their features are often presented in matrix form, each row of the matrix typically showing a distinct phone and its corresponding set of feature values. Some examples can be found in appendix B.

Although these feature classes and values are one of the most popular, there is no real agreement about exactly which ones should be used, hence several variants of this feature system exist in the literature (see for example Chang et al. [23], King et al. [81], Kirchhoff [83]). Another example is given in Table 4.2 (from Deng and Sun [33]). Feature classes are in this case the articulators of the human speech production system with different quantization levels.

Articulators	Values
Lips	0 to 5
Tongue blade	0 to 7
Tongue dorsum	0 to 20
Velum	1 to 2
Larynx	1 to 2

Table 4.2: Another example of phonetic (articulatory) features

As seen above, different feature systems exist. Their notion is also far from being new. Roman Jakobson was the first who introduced in 1941 the notion of distinctive features. In [71], Jakobson et al. proposed a set of such features based on auditory properties. Later in 1968, another breakthrough in this topic occurred with the articulatory approach published by Chomsky and Halle in *The Sound Pattern of English* (SPE) [25]. SPE features are similar to those found in Table 4.1, although they are different in two points: 1) features are not grouped in feature classes, 2) features have binary values (*e.g.*, '+' if the feature is detected, '-' if not). Chomsky and Halle defined in total twenty-two features applicable to any language, although not all features are relevant for all languages; only thirteen of them are used for English, as shown in Table 4.3. In the literature, some variants of these features also exist in order to adapt them to a specific corpus and its phone inventory (*e.g.*, Brondsted [17]). Among them, a *ternary* version can be used to mark non-relevant features (*e.g.*, voice for vowels) by another distinct symbol.

[1] It is said that the lower lip is the active articulator and the upper lip is the passive one, although it is less

Features	Values
Vocalic	+ or -
Consonantal	+ or -
High	+ or -
Back	+ or -
Low	+ or -
Anterior	+ or -
Coronal	+ or -
Round	+ or -
Voice	+ or -
Tense	+ or -
Continuant	+ or -
Nasal	+ or -
Strident	+ or -

Table 4.3: SPE features

4.3 Literature survey on phonetic features

This section will give a small survey of the literature about phonetic features. From the theories and experiments reported, some benefits of using these features will be brought to the fore.

A question one could ask indeed is: why a feature-based approach could be useful ? First of all, features are more fundamental units than phones and certain phenomena that seem complex at the phone level may be much simpler to describe at the articulatory level. King and Taylor [82] compares the phone-features relationship with atoms in physics: "If elements are described individually, they seem to exhibit idiosyncratic and somewhat arbitrary behavior. However, by describing them in terms of their sub-atomic makeup, the picture becomes much clearer (cf. the periodic table). The important point is that a small number of relatively simple sub-atomic particles can be used to describe the complex behavior of a much larger set of units from which they are made.". An example they give concerns phonotactics (well-formed phone sequences) in English. At the phonetic level, it is rather arduous to list all initial valid "CCC" (C = consonant) sequences of syllables (*e.g.*, s p l, s p r, s t r, s k r, s k w...) while at the feature level, simpler rules could be defined (*e.g.*, only voiceless stops can follow the initial [s] and only some laterals and approximants may follow the stops).

The same observation can be extended to pronunciation variation modeling. Stevens [132] gives an example with the word "ten": although its canonical pronunciation is [t eh n], some people with non-standard American English dialect tend to pronounce it [t ih n]. Although at the phone level a substitution occurred and the sound [eh] seems quite distinct from an [ih], there is actually not so much difference at the articulatory level: between the two sounds, only the tongue body has been raised to pronounce [ih] which corresponds to change only the feature *high* from '-' to '+' (in the SPE system). Eide [39] gives another example with the sequence "did you": in spontaneous speech, people tend to pronounce it [d ih jh uh] instead of [d ih d y uw]. Although substantial changes occurred at the phonetic level, only small variations are observed at the articulatory level: relative to the SPE system, only the features *anterior* and *strident* are modified in the collapsing of [d y] to [jh] and the feature "tense" for

clear here which one is which since both lips usually move.

the vowel substitution [uh] to [uw]. Because of its smaller variations, a feature-based approach may therefore provide an easier interface to capture pronunciation variations in spontaneous speech than a phone-based approach.

Another motivation to focus on phonetic features is due to an analysis made by Greenberg et al. [59] on Switchboard, a spontaneous telephone speech database [54]. They compared recognition and forced alignment outputs provided by eight speech recognition systems and analyzed them in order to point out which factors are the most important to take account in order to get a lower WER in spontaneous speech. About forty parameters pertaining to speaker, utterance, linguistic and acoustic domains were used for this purpose. Results show that performance improvement depends the most on accurate classification at both phonetic and articulatory levels. A more thorough analysis on phonetic features ([58]) revealed that there are three times more errors at the articulatory level when a word is misrecognized, suggesting again that taking account of phonetic features is important to deal with spontaneous speech. Another independent experiment made by Shinozaki and Furui [128] on some Japanese databases also revealed that accurate articulatory level classification was an important factor.

When a feature-based approach is adopted, the two following points need to be considered:

1. How can we obtain phonetic features ?
2. How can we incorporate features in ASR systems ?

Each question will be treated in detail in the next subsections.

4.3.1 How to obtain phonetic features

There are several ways to obtain phonetic (articulatory) features. The most reliable method is to directly measure trajectories from articulatory movements. Two equipments are commonly used for this purpose: the Electro-Magnetic Articulograph (EMA) [125] and the X-Ray MicroBeam (XRMB) [47]. An EMA measures movements of small sensors placed on the subject's speech production mechanism. Position and alignment of each sensor can be calculated from the currents induced by some transmitter coils positioned around the sensors and that generate an electro-magnetic field (a description of a 3D-EMA can be found in [162]). An XRMB-based system consists of sticking small gold pellets to human articulators and of tracking their trajectories using high energy x-ray microbeams directed towards the pellets. If one does not have any of these equipments, there is still the possibility to use some databases that already contain both speech and corresponding articulatory trajectories (*e.g.*, [146], [154]).

However, in practice, the databases with articulatory trajectories may not be suitable for the type of experiments one would like to carry out. Another alternative in this case is to infer the features from the acoustic data we are interested in. Methods for this purpose can be divided in two categories, either knowledge-based or data-driven. Knowledge-based methods consist of finding acoustic correlates of phonetic features in the speech signal, that is, of detecting and measuring some acoustic cues in the speech signal that give an indication about its articulatory properties. Stevens [132] proposed two types of acoustic cues: first for the *articulator-free* features, which indicate that some articulatory movements are noticed but do not specify which articulators are involved (*e.g.* "+consonantal" indicates that a constriction is formed in the oral cavity, but does not specify which articulators made the constriction), and second for the *articulator-bound* features, which specify the articulators (*e.g.*, "bilabial" implies that the lips made the constriction). The strategy adopted is first to find some landmarks

(*e.g.*, abrupt changes of amplitude) in the signal that approximately tell where to look in time to determine the articulator-free features, then to look more thoroughly around these landmarks (*e.g.*, formant movements) to determine the articulator-bound features. It was reported that combining several acoustic cues like this led to good discrimination between labial and alveolar stop consonants.

Bitar and Espy-Wilson [12] also defined a set of acoustic correlates of phonetic features. Types of acoustic correlates depend on the feature considered, they involve energy and autocorrelation coefficients of the signal, among others. A particularity of their method is that their parameters are relative in time and/or in frequency (*e.g.*, energy in a frequency band divided by the maximum energy in the same frequency band across the utterance) to implement the fact that the human auditory system relies on relative cues to phonetically identify speech segments, according to some psychoacoustic studies [64]. In [14], they used the Fisher criterion [36] to determine the parameters of their acoustic correlates (*e.g.*, frequency boundaries for an energy measure), then used a classification tree to prune redundant parameters. An average feature classification rate of 90.6% was obtained on their test set.

Although knowledge-based methods showed their usefulness, there are some requirements or drawbacks. First, some linguistic knowledge is required to define the acoustic correlates and to understand well their relationships with phonetic features. Second, knowledge about these relationships is so far incomplete: although related research is in progress, acoustic correlates are not yet fully accurate to detect all phonetic features reliably and we especially don't know much about their efficiency in spontaneous speech.

These reasons explain why data-driven methods have become popular. They consist in using pattern recognition tools that map acoustic parameters to phonetic feature values, trained with either real articulatory data (*e.g.*, using an EMA) or reference phone labels mapped to theoretical feature values (using phone-features conversion tables like those in appendix B) as targets. Several tools are available for this purpose, either deterministic or statistical. Suzuki et al. [137] built a codebook containing pairs of spectral segments and corresponding articulatory positions, trained from a database with speech and actually observed articulatory trajectories using an EMA. Articulatory parameters are then estimated by simply matching an input spectrum to those contained in the codebook and by extracting the corresponding articulatory parameters from the best pairs. A minimum square distance technique is finally applied to smooth the articulatory trajectories in time. They showed that estimated parameters using this technique are close to real ones (although with a same and single speaker enrolled for both training and testing). Kirchhoff [83] defined six feature classes (phonation, manner, place, front-back, roundness and centrality) and trained a left-to-right HMM with three to five states and single Gaussian per state for each feature value of each class. Data used to train the HMMs were standard MFCC acoustic vectors and training targets were obtained from manually labeled phones converted to phonetic features using a phone-features mapping table. For evaluation, HMMs of different classes were executed in parallel in order to define the feature value of each class. An average of 91.8% feature recognition rate was achieved. Koreman et al. [90] used Kohonen networks[2] to map acoustic vectors to phonetic features for each frame. The final features for a given frame are obtained by using a weighted average of the output values associated with the K winning neurons closest to the input acoustic vector. No accuracy result of the acoustic-to-feature mapping was reported. Stephenson et

[2]Kohonen networks consist of neurons distributed in two or three dimensions representing a certain source domain (*e.g.*, cepstral domain in our context). These neurons have the particularity of being ordered in space. After stimulation and calibration of the networks, each neuron is associated to both the location of a distinct centroid in the source domain and a vector that maps the source to an (averaged) destination value. More detailed information can be found in Kohonen [88] and Dalsgaard [32].

al. [130] trained dynamic Bayesian networks[3] (DBNs) from real articulatory data to infer the articulatory positions from the acoustics. Although accuracy of the acoustic-to-articulatory mapping was not reported, the small difference in WER obtained between two systems with real articulatory data on one hand and inferred by the DBNs on the other hand suggests that the mapping was fairly accurate.

The most popular pattern recognition tool to map acoustic data to phonetic features is probably neural networks (*e.g.*, Chang et al. [22], King and Taylor [82], Kirchhoff [86], Papcun et al. [110], Zacks and Thomas [161]). For example, King and Taylor [82] trained neural networks with single hidden layer to map MFCC parameters to phonetic features. Training targets were manually labeled phones mapped to phonetic features using a phone-features conversion table. Three feature systems were compared: the Multi-Valued system (cf. Table 4.1 for an example), the SPE system (cf. Table 4.3) and the Government Phonology (system based on combinations of primes, cf. [62]). All three systems led to high percentages of frames correct for each feature considered separately (average percentages were between 86% and 93%).

4.3.2 How to incorporate phonetic features in ASR systems

There are several ways to incorporate phonetic (articulatory) features in ASR systems. The most common and simplest way is to use the feature values (*e.g.*, activation values of output neurons when using neural networks to map acoustic vectors to phonetic features) instead of the usual acoustic parameters (*e.g.*, MFCC) to train HMMs. King and Taylor [82] used their phonetic features to train cross-word triphone models and evaluated them with phone recognition; similar performance to MFCCs were obtained. Koreman et al. [90] did similarly with phonetic feature strengths obtained from their Kohonen networks. They obtained substantial improvement in identifying consonants and their places of articulation using this method, although from a low baseline since they did not use any language model or lexicon. Extension of the recognition to all phones [89] with two different feature systems led to similar results with single Gaussian per HMM state; however, phonetic features performed less well than MFCCs when the number of Gaussian mixtures per state was increased to eight. It was also reported that confusions between phones were less severe with phonetic features than with acoustic parameters when using a single Gaussian per HMM state, but again this benefit disappeared when the number of Gaussian mixtures was increased. Bitar and Espy-Wilson [13] defined some acoustic correlates of phonetic features and used them instead of MFCCs to train and evaluate HMMs for broad phonetic class recognition. Comparable results to MFCCs were obtained with 8 Gaussian mixtures per state. Furthermore, they experimentally showed that their new parameters were more robust to gender variability because they are based on relative measures (*e.g.*, division of two energy measures). The approach of Dalsgaard [32] is a bit different from the previous authors in that feature strengths obtained from Kohonen networks were used to build separate histograms for each feature of each phoneme. Then, a Gaussian probability density function was approximated for each histogram. Finally, phoneme models were built from mixtures of these approximating Gaussians. A Viterbi alignment process to define phoneme label alignment resulted in 85% correct with Danish, however only 43% correct with British English (due to insufficient training data according to the author).

[3]Bayesian networks consist of oriented and acyclic graphs where each node is associated with a specific variable (*e.g.*, state, acoustic observation) and a conditional probability for the variable (*e.g.*, transition or emission probabilities when compared to HMMs). The dynamic version extends the initial possibilities by including dynamic processes like time. Bayesian networks are sometimes used as alternatives to HMMs for ASR. More information can be found in Zweig [163].

Some papers reported that acoustic and articulatory information are partially complementary and they often do not generate the same types of errors in recognition (*e.g.*, [84]), hence they advocate that they should be combined. A question relative to this is whether all acoustic and phonetic feature parameters should be combined, and if not, how to select an optimal subset. Retaining more feature parameters would a priori help to improve the accuracy of the resulting acoustic models, but provided there is enough training data to estimate the parameters and with the cost of increasing the model complexity. Kirchhoff [86] designed a backward discriminative feature selection algorithm: starting with 65 combined acoustic and phonetic feature parameters, each feature was hypothetically removed and the one that led, with the remaining features, to the biggest average difference in log-likelihood between the correct model and incorrect models (correct model was given by reference labels and segmentations) was eliminated. Another iteration was then applied with the remaining features, and so on until the number of desired features was obtained. Experiments showed that the combination scheme led to higher improvement compared to the replacement scheme with spontaneous speech, although this improvement was still rather marginal compared to the baseline system with only MFCCs. Eide [39] started to concatenate all SPE features to acoustic parameters, but then computed mutual information between the estimated and true feature values. Only four features with the highest mutual information values were retained and appended to MFCCs. Substantial improvement was obtained in three different car noise conditions (contribution of phonetic features in robustness to noise was also noticed by Kirchhoff [85] with two different types of noise at multiple signal-to-noise ratios).

Acoustic and articulatory information can also be combined at other levels of an ASR system. Kirchhoff [85] evaluated combination schemes also at the state and word levels. At the state level, outputs of two neural networks - one that maps MFCCs to phones and the other that maps phonetic feature strengths to phones - were combined using a weighted sum or product rule to obtain the posterior probability of a phone given the two types of parameters. These probabilities were then fed into a hybrid ANN/HMM system for word recognition. Significant improvements relative to the MFCC baseline were obtained. At the word level, the best output word sequences of two ASR systems respectively based on acoustic and articulatory parameters were combined in an N-best list rescoring scheme. Frequencies of word hypotheses and word confidence values were used to rescore the lattices of hypotheses. Again, substantial improvement over the MFCC baseline was obtained. Other examples of combination of acoustic and articulatory information will be mentioned in the remainder of this section.

Phonetic features can also govern acoustic model topologies. Several reasons generally motivate the design of such techniques:

1. Current standard models like HMMs model speech, but without taking account of the speech production mechanism; many researchers think that incorporation of articulatory knowledge might be beneficiary for speech recognition (this is a reason also valid for the previous seen techniques).

2. Since parts of the vocal tract are mostly independent of each other, they do not generally move simultaneously during speech production and transitions are not instantaneous. A more sophisticated topology than the well-known three left-to-right HMM states is therefore desirable to model these asynchronous transitions more accurately.

3. Current HMM-based ASR systems generally model coarticulations by simply modeling all phonetic contexts of all phones (at least those seen in the database). This is a

rather blind method and may lead to data scarcity problems, especially in large vocabulary speech recognition where the number of possible contexts is substantially higher. Clustering techniques are generally used to overcome this situation. It is known however that coarticulation phenomena are associated with the movements of articulators ([61]). Modeling of coarticulations could therefore be guided by the theories of speech production and data could be shared more intelligibly and parsimoniously.

The most thorough work in this topic has probably been done by Deng and his colleagues [33]. They chose five articulators (represented in Table 4.2) and associated each context-independent phone with a specific combination of articulatory positions. To model coarticulations, articulatory features of two successive phones were allowed to *overlap* each other. Consequently, each context-dependent phone was modeled by a different HMM topology, with states representing the transitions of articulators influenced by the neighbor phones. Asynchronous transitions were modeled by parallel paths in the topology, in which one articulator moved before another. Experiment results showed that phonetic classification using this method outperformed standard context-independent phonemic HMMs. They were comparable in performance to context-dependent models, however with much less training data. In [136], a data-driven version based on regression trees is proposed to construct the initially rule-based HMM topologies. A 7.2% relative improvement in phone recognition accuracy was observed on the TIMIT database. Richardson et al. [116] followed a similar line of thought and defined a specific articulator-based HMM (so-called Hidden Articulator Markov Model, HAMM) topology for each diphone, with static and dynamic constraints to limit the number of possible articulatory configurations and movements. Although the HAMMs alone did not improve performance over their baseline HMMs, a weighted combination of log-likelihoods from both systems led to significant improvement. Furthermore, they showed in [117] that the same combination was also more robust to noise than their baseline system alone. Frankel and King [46] modeled articulatory trajectories using linear dynamic models (LDM) instead of HMMs, described by two equations: (1) $x_t = Fx_{t-1} + \eta_t$ (η_t: Gaussian distribution noise) describes how the state variable x_t varies from frame to frame and (2) $y_t = Hx_t + \epsilon_t$ (ϵ_t: Gaussian distribution noise) associates the state with an observation vector y_t. This representation has the benefit of modeling states (interpreted again as articulatory positions) in a continuous space. Although use of articulatory data alone was not successful, LDMs trained with both acoustic and real articulatory data yielded higher phonetic classification accuracy. Stephenson et al. [130] added new nodes and arcs in their Bayesian networks to represent articulatory positions, which correspond to take account of an articulator variable a_t in emission and transition probabilities: $P(x_t|a_t, q_t)$ and $P(a_t|a_{t-1}, q_t)$, respectively (x_t: acoustic vector, q_t: state). Related experiments led to about 10% relative reduction in WER compared to their baseline without articulatory information.

Finally, phonetic features can also be used to access the lexicon. Kirchhoff [83] stored syllables in lexicon as parallel sequences of phonetic features (one sequence per feature class). Syllable recognition was performed by comparing the same type of sequences automatically extracted from speech data to the corresponding sequences of each syllable template in the lexicon using dynamic programming. Although syllable-level recognition performance was not compared to any baseline, recognition accuracy of phonemes derived from top syllable sequences outperformed a standard triphone-based recognition system. Lahiri [95] created a lexicon with word entries represented by sequences of *underspecified* phonetic features, that is, some features were not specified because their values could vary due to pronunciation variations. For example, the word "green" is basically pronounced [g r iy n], but can also be pronounced [g r iy m] before a bilabial consonant (like in "gree[m] bag") due to the coar-

ticulation involved by the spread of the bilabial feature of [b] to [n]. Since [n] and [m] have different places of articulation, the corresponding feature is voluntarily not specified. Lexicon access was performed by matching phonetic features extracted from the input speech to those stored in the lexicon, with three possible results, "match", "no mismatch" and "mismatch" ("no mismatch" typically referred to underspecified features). Word candidates with "match" or "no mismatch" were retained for further process by a phonological and syntactic parser. No experiment result was reported.

4.3.3 Survey summary, benefits and issues

In the previous subsections, some examples of applications and benefits of phonetic features were presented. For the sake of bringing to the fore the essential ideas, the most important points will be listed below with the relevant references.

In order to build an ASR system based on phonetic features, two points need to be considered: how to obtain the features and how to incorporate them into ASR systems. Features can be obtained using one of the following methods:

- Usage of an electro-magnetic articulograph [125] or an X-ray microbeam [47] to measure articulatory trajectories during speech production, or alternatively usage of a database that already contains such trajectories (Westbury et al. [146], Wrench [154])
- Detection and measurement of some acoustic cues in the speech signal that give an indication about its articulatory properties (Bitar and Espy-Wilson [12], Stevens [132])
- Usage of deterministic or statistical tools to infer the features from the acoustics:
  - codebooks (Suzuki et al. [137])
  - HMMs (Kirchhoff [83])
  - Kohonen networks (Dalsgaard [32], Koreman et al. [90])
  - Bayesian networks (Stephenson et al. [130])
  - neural networks (*e.g.*, King and Taylor [82])

Methods to incorporate phonetic features into ASR systems include the following techniques:

- Replacement of acoustic parameters (*e.g.*, MFCCs) by feature values to train and recognize with HMMs (Bitar and Espy-Wilson [13], King and Taylor [82], Koreman and Andreeva [89])
- Combination of acoustic and articulatory information at feature (Eide [39], Kirchhoff [86]), state or word levels (Kirchhoff [85])
- Design of acoustic model topologies to reflect articulatory positions and movements (Deng and Sun [33], Frankel and King [46], Richardson et al. [116], Stephenson et al. [130])
- Usage of features to access the recognition lexicon (Kirchhoff [83], Lahiri [95])

Some motivations and benefits (real or potential) of using phonetic features are as follows:

- Convenient interface to describe complex rules and pronunciation variations (Eide [39], King and Taylor [82], Stevens [132])
- Important factor for better recognition performance in spontaneous speech (Greenberg et al. [58] [59], Shinozaki and Furui [128])
- More acceptable phonetic confusions (Koreman et al. [90])
- Better management of coarticulations (Deng and Sun [33], Richardson et al. [116])
- Less training data required (Deng and Sun [33])
- More robust to noise when combined with acoustic parameters (Kirchhoff [85], Richardson et al. [117])

The last but not the least benefit associated with phonetic features is that they are not bound to any language; a feature-based approach is therefore suitable for *any* language (at least in theory). An example towards this concept is given by Chang et al. [22] and Wester et al. [148] who mapped acoustic parameters to phonetic features using an American English and Dutch database, respectively. Because the symbol sets of both languages were described by the same set of features, a cross-linguistic classification could easily be performed: neural networks were trained with the American English database, but evaluated with the Dutch database. Although some feature classes (place of articulation) suffered a substantial degradation, some other classes (voicing and manner of articulation) transferred fairly well. Furthermore, improvements obtained in one language were also observed in the other language.

Despite the firm conviction by many authors that use of phonetic features can be beneficial for speech recognition, there is still a major issue that needs to be addressed: most experiments reported in published papers concerned isolated word recognition or read speech recognition (*e.g.*, TIMIT). Such experiments are surely useful as an initial step, but it still needs to be shown that the same methods would be as successful under more difficult conditions (*e.g.*, spontaneous speech, large vocabulary). Some authors reported some small improvement in spontaneous speech (*e.g.*, Wester et al. [148]), but concerned phone-level and not word-level recognition. Kirchhoff [86] did compare word error rates using phonetic features with both spontaneous and large vocabulary speech recognition, but no gain could be obtained compared to MFCC-based systems. Other authors (*e.g.*, [60]) also expressed their difficulty to apply their methods to spontaneous speech. More time and study are therefore needed to address this issue, but all the successful attempts achieved on simpler tasks predict some encouraging results in the future.

4.4 Introduction to the applied methodology

The method applied in this chapter is inspired from some of the techniques seen in the literature survey, but adapted for pronunciation variation modeling. Recalling what was mentioned in the general overview of section 4.1, the idea put forward was to build a dynamic lexicon, that is, a lexicon who is able to adapt its content to the pronunciation characteristics of an input speech. For this purpose, the two following parts were considered:

1. A *static augmented lexicon* building part: it consists of discovering new pronunciation variants for each word and of adding them to the basic lexicon.

2. A *dynamic lexicon* building part: it consists, during recognition, of selecting among the entries available in the static augmented lexicon the phonetic transcriptions that best match the pronunciation characteristics of an utterance. A smaller lexicon with only these transcriptions is consequently generated.

Each step will be described in detail in the following sections.

4.5 Static augmented lexicon building

The objective of this first part is to automatically discover possible relevant transcriptions for each word, in order to generate a lexicon with new pronunciation variants. A first step consists of generating all possible transcriptions given an utterance, followed by a selection step that chooses the most likely variants among the ones proposed.

Let us first consider the generation step. We already expressed our interest to include articulatory knowledge to our ASR system. Recalling some of the benefits already mentioned, a feature-based approach could be an easier interface to capture pronunciation variations and it could generate more acceptable phonetic confusions. Relative to the generation of pronunciation variants, the latter means that even if detection of features does not accurately target true pronunciations, the resulting errors may be more acceptable phonetically to be considered as alternative pronunciations. A feature-based approach could therefore be more convenient to generate pronunciation networks than a phone-based approach.

A feature-based technique was therefore adopted for the generation step. Several decisions relative to this choice had then to be made:

1. Which features to use ?
2. How to extract the features from the input speech ?
3. How to generate new pronunciation variants from the features ?

Exactly which feature system to use did not matter much a priori since several systems were said to yield similar performance (King and Taylor [82]). Our choice was simply based on popularity of the system and its simplicity of implementation. The SPE system (Chomsky and Halle [25], features are listed in Table 4.3) is popular and is convenient to use because all its features are binary and organized in a single class, hence it was the set chosen for our experiments.

Concerning the extraction of features, it was already mentioned that knowledge-based methods based on acoustic correlates of phonetic features were so far incomplete. Furthermore, several experiments showed that phonetic features could automatically be extracted from speech using pattern recognition tools with fairly good results (*e.g.*, Chang et al. [22]), which naturally influenced our choice in favor of data-driven methods. It remained to decide for a system to map acoustic to articulatory parameters. According to several studies in this topic (*e.g.*, Atal et al. [6], Bailly et al. [8]), extraction of articulatory information from acoustics is commonly known as an *inverse mapping* or *many-to-one* problem, in that many different articulatory configurations may produce the same acoustic patterns. Stevens wrote indeed in [131] that the acoustic-articulatory relationships are *quantal*, as shown by the schematization in Figure 4.2: three zones are defined, and while the articulatory parameter smoothly changes from zone I to III, the acoustic parameter value almost jumps from one state to another in

zone II and remains relatively stable in the other zones, so similar acoustic state for different articulatory configurations.

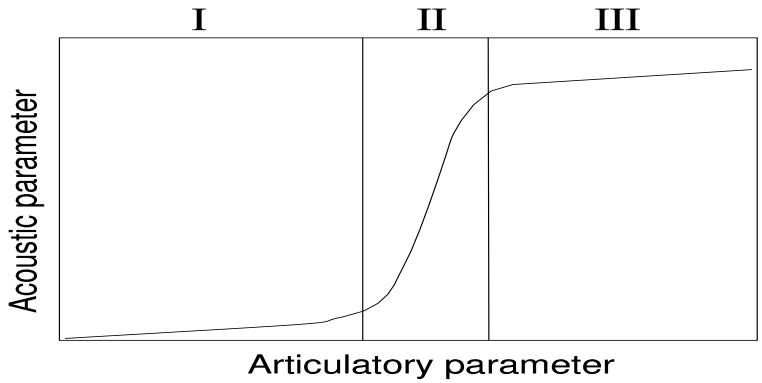

Figure 4.2: Schematization of a quantal relationship between acoustic and articulatory parameters (from Stevens [131])

Although it was not certified that this type of relation was valid for all features, an experiment carried out by Gay et al. [52] also showed that people were able to utter vowels with similar acoustic patterns even though their articulatory system was disturbed by some obstructions. Acoustic-to-articulatory mapping is therefore not unique and highly non-linear and cannot be reliably estimated by deterministic methods. Among the available choices of statistical-based techniques, neural networks were reported by certain authors as an appropriate tool (*e.g.*, King et al. [81], Kirchhoff [85]) - which also influenced our choice - for the following reasons:

- Neural networks are flexible to model any non-linear mapping, provided that enough free parameters and data to reliably estimate them are available. In particular, they do not assume any distribution like HMMs. It is true however that rather complex distributions can also be modeled by mixtures of Gaussians. The best choice depends therefore on the complexity of the model required to accurately approximate the target distribution.

- Acoustic-to-articulatory inversion mapping will be more robust if a longer time interval is considered, since articulatory properties are often not restricted to a single phonetic unit but tend to be spread to neighbor phonetic contexts. Neural networks easily provide this possibility and can accept any number of time frames as input.

- Training of neural networks is discriminative, that is, it tends to mark the differences and boundaries between the possible output classes rather than tries to best characterize the distribution of the speech signal by maximizing its likelihood. Standard training algorithms for HMMs (*e.g.*, Baum-Welch) do not have this property, although discriminative training methods also exist (*e.g.*, Chesta et al. [24], Juang et al. [74]).

- It was reported that neural networks are more accurate than HMMs to perform acoustic-to-articulatory mapping (see King et al. [81]).

- Neural networks are the most often used in the literature to extract phonetic features from speech and are freely available in the Internet (*e.g.*, [107]).

Based on the choices above, the following general procedure was applied to discover new pronunciation variants for each training utterance (Figure 4.3):

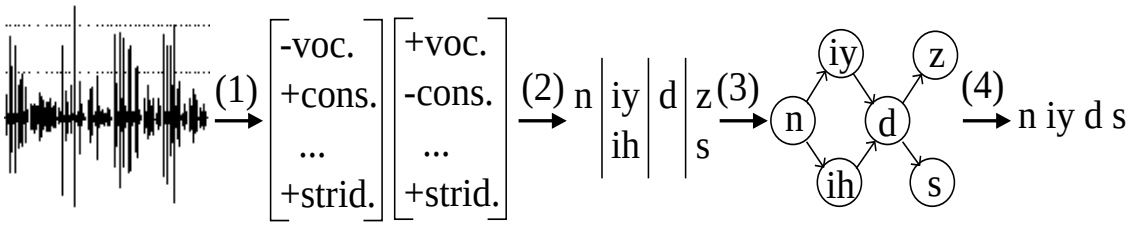

Figure 4.3: Steps to build a static augmented lexicon

1. Some phonetic features were first extracted from the input speech on a frame-by-frame basis using an artificial neural network (ANN). A paradigm based on output activation values was adopted to search also for alternative combinations of features per frame.

2. Each combination of detected features for a given frame was mapped to a phone using a phone-features conversion table.

3. Successive frames mapped to a same phone were grouped to form hypotheses, which were then connected to each other to build a pronunciation network.

4. All possible phonetic transcriptions were generated from the network and the most likely ones were selected by means of a two-pass Viterbi alignment and a pruning process.

The static augmented lexicon was obtained by adding all the selected transcriptions to the basic lexicon. The following subsections will describe each step in detail.

4.5.1 From speech to phonetic features

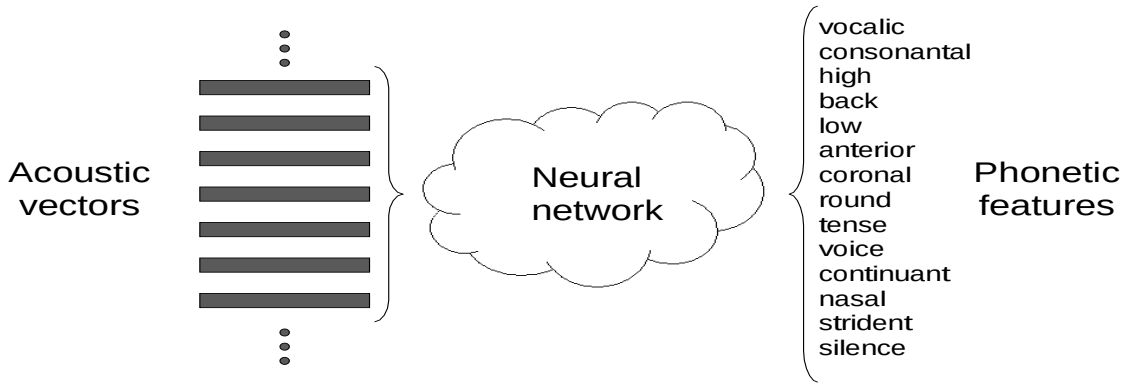

Figure 4.4: Acoustic-to-articulatory mapping

An artificial neural network (ANN) mapped a set of acoustic parameters to phonetic features (Figure 4.4). The net had the three basic layers: one input, one hidden and one output. Each node of the input layer accepted an acoustic parameter (Mel-frequency cepstral coefficient or normalized log energy in our experiments) and each node of the output layer gave the strength of a particular feature (SPE features in our experiments). Inclusion of context frames as inputs of an ANN is known to be important for significantly better detection of phonetic features (see *e.g.*, Kirchhoff [85], chapter 3). Between 5 and 9 frames are generally used as inputs for phonetic feature recognition in the literature. Our own experiments used 7 context frames. Time derivatives of acoustic parameters were also taken into account by the

ANN architecture. Let us consider the network as a black box for the moment, more detail will be given in the experiments section (4.7.3).

The acoustic-to-articulatory mapping was done on a frame-by-frame basis. For training simplicity, features were considered as independent; the ANN performed therefore an N-to-M classification, that is, several output nodes could be activated simultaneously. The tangent hyperbolic was used as the activation function, so each feature for which the corresponding output neuron displayed a positive activation value was considered as detected, although under some additional conditions (see the next subsection).

4.5.2 From phonetic features to alternative feature combinations

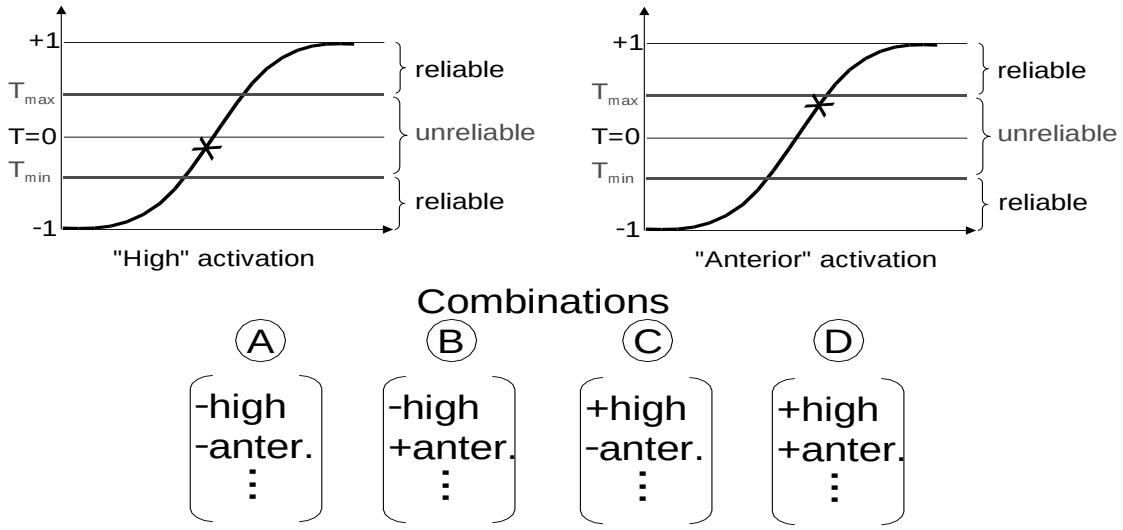

Figure 4.5: Process to generate alternative feature combinations

From the previous step, phonetic feature strengths were estimated from speech and the threshold applied to the activation values of output neurons decided whether the feature was present or not in the signal at the given frame. The combination of these features corresponded to a unique sound. However, with the objective of discovering pronunciation variants, it was desirable to consider also some feature alternatives, so the following paradigm was adopted. Assuming that the acoustic-to-articulatory mapping performed by the ANN was fairly but not fully reliable, an incertitude interval was set around the activation threshold of each output neuron. Any feature value who fell inside this interval was classified as *unreliable*, and both presence and absence of this feature were considered. An example is given in Figure 4.5. The curves in the figure represent the activation functions of the output neurons associated with the features "high" and "anterior". Around the activation threshold T of each function, an incertitude interval given by T_{min} and T_{max} was defined. In this example and for a given frame, both feature values fell in this interval. Consequently, all possible values of these two features were considered for this frame while the other features remained unchanged (assuming they were outside the incertitude interval). This resulted in four possible combinations as shown in the figure.

It is clear that the number of possible combinations rapidly increased with the number of unreliable features. To prevent an explosion of possibilities, combinations were ranked so that only a subset of them could be kept in case of too many alternatives. Two ranking criteria were adopted. First and foremost, combinations with original feature values were favored. In the

example of Figure 4.5, "high" and "anterior" were originally set as "-high" and "+anterior" respectively by the ANN. Hence, the combination "B" was the most preferred, since both features kept the original values. The second criterion was the distance of the original feature value from the activation threshold T: the closer was a feature value from this threshold, the more uncertain we were about the feature presence or absence, hence the more plausible it was to test both possibilities. In the example mentioned previously, combination "D" was favored to "A" because the "high" feature value was originally closer to the activation threshold T (hence more likely to be modified) than "anterior". An explicit *penalty score* was associated with each combination to quantify these preferences and was given by the sum of activation distances to T of phonetic features modified from their original states. For example, supposing that "high" had an activation value of -0.1 (feature absent) and "anterior" had +0.4 (feature present), any combination that required "high" to be active underwent a penalty of 0.1, and 0.4 if it considered "anterior" as absent. According to the two criteria and penalty scores mentioned, the four combinations were ranked in the following order with the corresponding penalty scores (in parentheses): B (0.0), D (0.1), A (0.4), C (0.5). This ranking process was used during the *generation* of pronunciation variants (building of pronunciation network), but did not influence the *selection* of the best variants (see section 4.5.6). Besides, penalty scores were also used to evaluate the accuracy of pronunciation networks generated with the procedure being explained (see section 4.8.2).

4.5.3 From feature combinations to phones

Alternative combinations of phonetic features listed some articulatory configurations that could produce the acoustics presented at the input. As advocated by linguists, phonetic features should normally be used directly to format entries of the ASR lexicon and to express pronunciation variations directly from the articulatory point of view. However, for practical reasons, we still decided to map feature combinations to phones. The main reason was that a direct utilization of feature combinations in the ASR lexicon would have implied estimation of a separate acoustic model (HMM) for each combination (recalling that our initial objective was to improve performance by modifying the lexicon, but while still using HMMs). However, the number of possible feature combinations was fairly high (more than 16000 possibilities with SPE features), which would have led to data scarcity problems and required some model clustering techniques. As long as HMM topology, training algorithm and clustering techniques were standard, resulting models would a priori not have been much different from context-dependent phone-based models (triphones referring to a same monophone may correspond to similar articulatory configurations, but with slight feature differences between them). For the sake of a fairer comparison with our baseline system (not based on phonetic features), we preferred to keep the phone-based models. A more interesting approach in our context would have been to keep using a phone-based representation in the lexicon, but to adapt their HMM topologies based on the phonetic feature combinations extracted from speech; in other words, to adopt a strategy similar to the idea of Deng and Sun [33]. But this technique goes beyond the scope of lexicon level pronunciation modeling initiated in this chapter and therefore will not be considered. Nevertheless, another data-driven method that modifies HMM topologies in pronunciation variation modeling perspective will be presented in the next chapter.

Therefore, each group of phonetic features was directly mapped to a phone using a phone-features conversion table (see appendix B.1). By doing this for all combinations of all frames, each frame was associated with one or more phones. However, some combinations could not be matched to any phone listed in the table. These cases are explained by the asynchronism of articulatory movements: parts of the vocal tract do not move simultaneously and instan-

taneously, but often move from one state to another at different time intervals and speeds. Coarticulation also contributes to this asynchronism, so that features of neighbor phonetic units overlap each other. A feature value located in the incertitude zone by the ANN could therefore be seen as a transition state in the vicinity of two adjacent phonetic segments with opposite values for this feature. Following this point of view, frames for which such case occurred were called *transitional frames* and the corresponding combination of features were mapped to a so-called *transitional phone* (*i.e.*, a virtual phone not listed in the phone inventory and not modeled with HMMs). An example following the case illustrated in the previous step (Figure 4.5) is shown in Figure 4.6. Among the four possible combinations generated by changing the values of "high" and "anterior" features, two of them were successfully mapped to valid phones [s] and [sh], while the remaining two could not be found in the phone-features conversion table and hence were simply mapped to a transitional phone [?]. All mapped phones conserved the same preference order as with feature combinations, except if there were more than one transitional phone in the list, in which case only the best transitional phone was kept and the others were discarded. The concept of transitional phones and frames were used in two ways: first, they helped decide whether a phone segment hypothesis built from mapped phones was reliable or not (see section 4.5.4); second, they helped to score pronunciations during the construction of dynamic lexicons (see section 4.6.3).

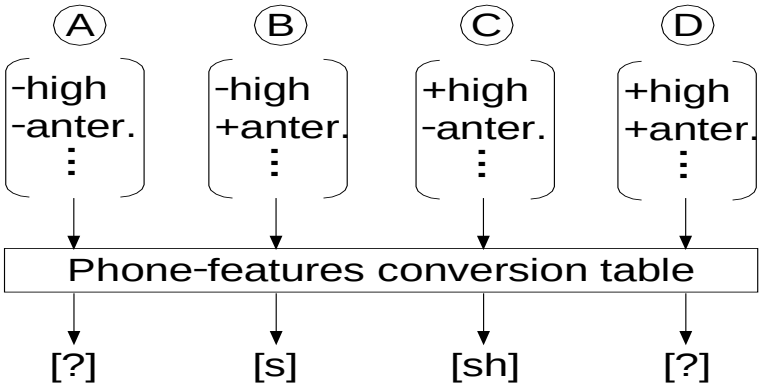

Figure 4.6: Mapping of feature combinations to phones

4.5.4 From phones to phone segment hypotheses

At the end of the previous step, each frame was associated with a ranked list of phonetic feature combinations mapped in turn to a ranked list of non-transitional (called *valid* hereafter) or transitional phones. In order to build a pronunciation network, we first built the nodes of the network: a phone segment hypothesis was created whenever F_{min} ($F_{min} = 3$ in our experiments) or more successive frames referred to a same phone in their respective lists. Moreover, a hypothesis was considered as *reliable* if at least R_{min} ($R_{min} \leq F_{min}$, $R_{min} = 2$ in our experiments) of its frames respected the following conditions:

1. Only one valid phone was associated with the frame.
2. No transitional phone was ranked better than the valid phone.

Reliability of hypotheses intervened during the pronunciation network building (see section 4.5.5). As a frame could be associated with more than a single phone, hypotheses could overlap partially or even totally in time, as shown by the example in Figure 4.7 for the sequence of words "he needs to go".

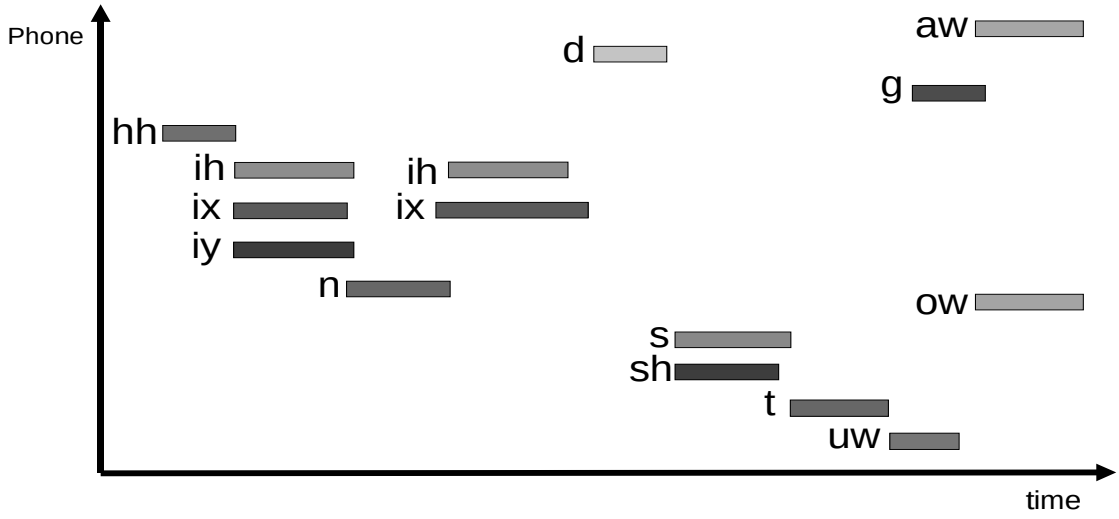

Figure 4.7: Example of phone segment hypotheses built from phones

4.5.5 From phone segment hypotheses to pronunciation network

Since phone segment hypotheses represented the nodes of a pronunciation network, it remained to link them to each other. Some constraints checked whether two hypotheses were not too far away to be connected, and then tested for possible succession or substitution relationships between them based on how much they overlapped. The general algorithm is given below:

```
Initialization:  sort hypotheses in increasing order of their starting time

Loop A: for each hypothesis A in the ordered list

   Loop B: for each hypothesis B ordered after hypothesis A in the list

      If A and B are too distant from each other, go to ‘‘End Loop B’’

      If B can follow A
         Create a link from A to B

      Else if A can follow B
         Create a link from B to A

   End Loop B

   If A is not the last node and has no successor
      Create a link from A to the nearest hypothesis starting after end of A

End Loop A
```

Creation of a link between two hypotheses implies a succession relationship between them, so the algorithm simply tests whether a hypothesis can directly follow another. If it is the case, a connection is created to join the hypotheses. Otherwise, nothing is done because either the hypotheses are too far away from each other or they represent alternative paths in the pronunciation network. Hypotheses were considered too distant from each other when the number of frames separating the hypotheses was equal or bigger than F_{min}, defined previously as the minimum number of frames required to create a valid hypothesis; it means that a gap equal or bigger than this value may allow insertion of one or more hypotheses in between, hence the connection should not be made. The following points define the conditions required

for a hypothesis B to follow a hypothesis A . These conditions were only based on relative positions of hypotheses in time; they were heuristically defined and certainly leave some room for improvement, but still provided some reasonable results. Each point is illustrated in Figure 4.8:

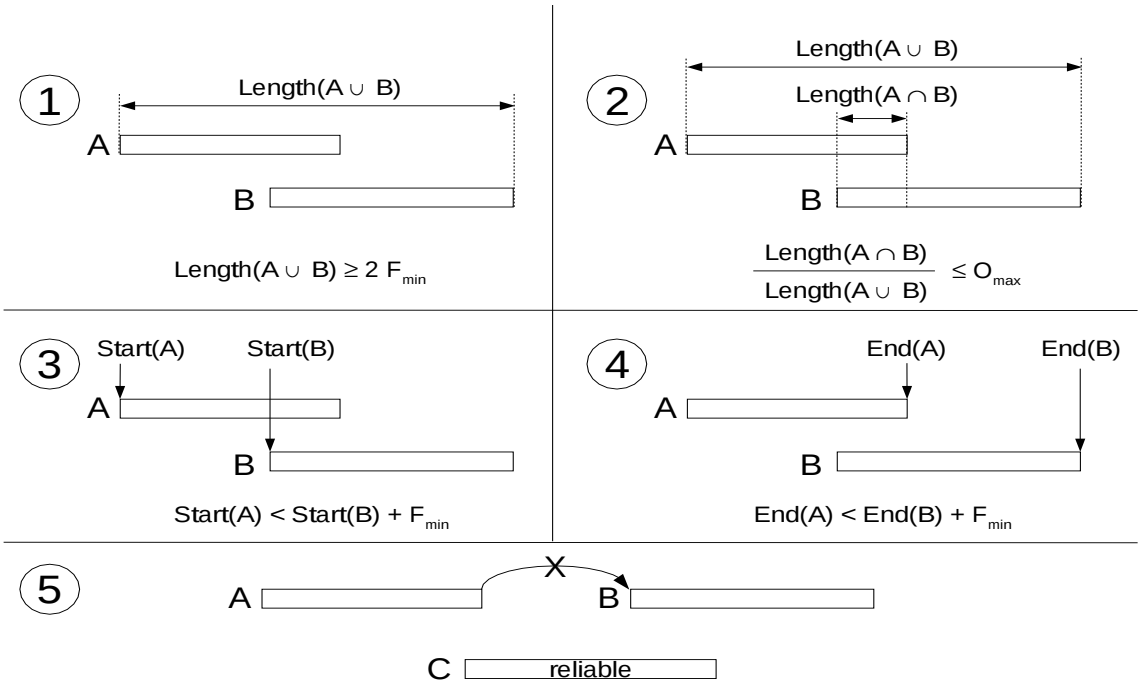

Figure 4.8: Conditions to connect two hypotheses A and B

1. $Length(A \cup B) \geq 2F_{min}$: even if A and B overlap each other, the length resulting from their union must be long enough to hold at least two hypotheses.

2. $Overlap(A, B) < O_{max}$: A and B must not overlap more than a certain ratio O_{max} (fixed to 50% in our experiments). An overlap rate was simply defined by $\frac{Length(A \cap B)}{Length(A \cup B)}$, that is, the number of frames associated to both A and B divided by the number of frames resulting from their union.

3. $Start(A) < Start(B) + F_{min}$: A must start before B or at least not too late after B. If A starts F_{min} or more frames after B, A cannot precede B. A tolerance interval of $F_{min} - 1$ frames was left at the right of B to take account of uncertainties to locate the exact starting points of hypotheses due to transitional frames.

4. $End(A) < End(B) + F_{min}$: A must end before B or at least not F_{min} frames or more after B. The same tolerance interval as for the previous condition applies.

5. $\nexists C \,|\, C\, reliable,\ Start(A) \leq Start(C) \leq Start(B)$: if there is a reliable hypothesis C (notion of reliability was defined in section 4.5.4) between A and B, it cannot be skipped.

Once the pronunciation network was built, each single path through the network represented a possible transcription. To get pronunciation variants on a per word basis, the whole network was finally segmented into subnetworks (one per word) according to some manually defined word-level time boundaries. An example of possible pronunciation network and corresponding word-level segmentation is shown in Figure 4.9. At this stage of process, some words

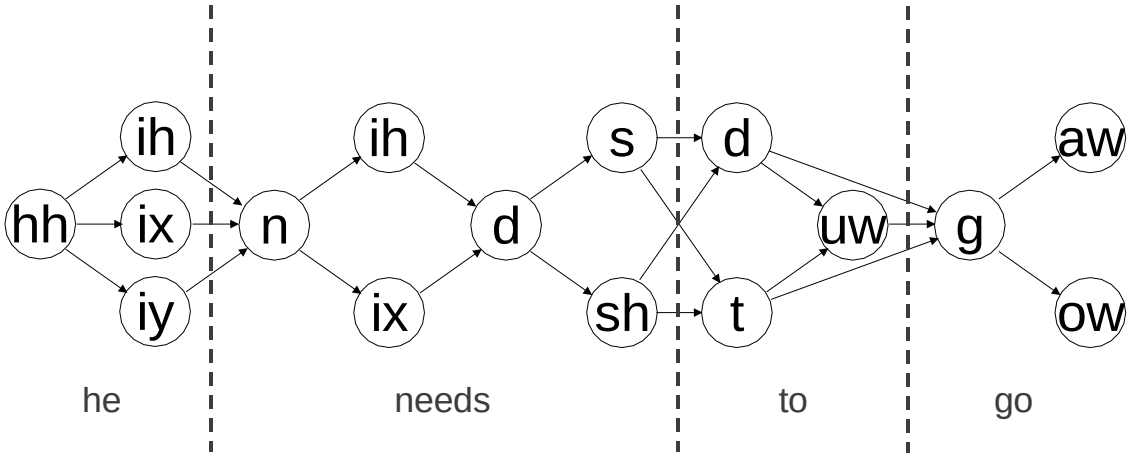

Figure 4.9: Example of pronunciation network built from phone segment hypotheses

did not have the baseform transcription available (*e.g.*, "needs" with canonical transcription /n iy d z/ in this figure). To insure that pronunciation variants always competed with the corresponding canonical pronunciation during the selection process (cf. section 4.5.6), the baseform transcription of each word was added to the network whenever it was not available. Hence for the word "needs" in the example, a separate path representing /n iy d z/ would be added.

4.5.6 Selection of pronunciation variants

All the previous steps described how to generate a pronunciation network from the input speech through the detection of phonetic features. The networks yielded too many transcriptions to be added to the basic lexicon without a substantial increase in lexical confusability, so a selection process was necessary to keep only those that best matched the input data. For this purpose, Viterbi alignment was applied to select the best paths (according to the maximum likelihood criterion) in the networks. Actually, two passes of Viterbi alignment were performed on the training data. The first pass consisted in selecting the best transcriptions among those provided by the pronunciation networks. Then, all transcriptions selected at least once by the first pass were made available during the second pass that further restricted the number of available variants: for each word of a training utterance, all phonetic transcriptions referring to this word and selected during any utterance of the first pass competed with each other. Only phonetic transcriptions selected at least once by each of these two passes were added to the basic lexicon.

An example is illustrated in Figure 4.10. To simplify the illustration, let us assume that the training database contains only two utterances (sentences) with the word "needs". The generation process described previously creates a specific pronunciation network for each utterance. During the first pass, a Viterbi alignment is applied on each network. In this example, the pronunciation variants [n ih d s] and [n ih t z] were selected for utterances 1 and 2, respectively. Next, these two transcriptions compete with each other in the second pass using the same two utterances. According to the example, only [n ih d s] was preferred for both utterances and hence is the only pronunciation variant finally added to the basic lexicon.

Furthermore, the most frequent words (typically function words such as "and") still led to too many pronunciation variants despite the two Viterbi alignment passes. Selected transcrip-

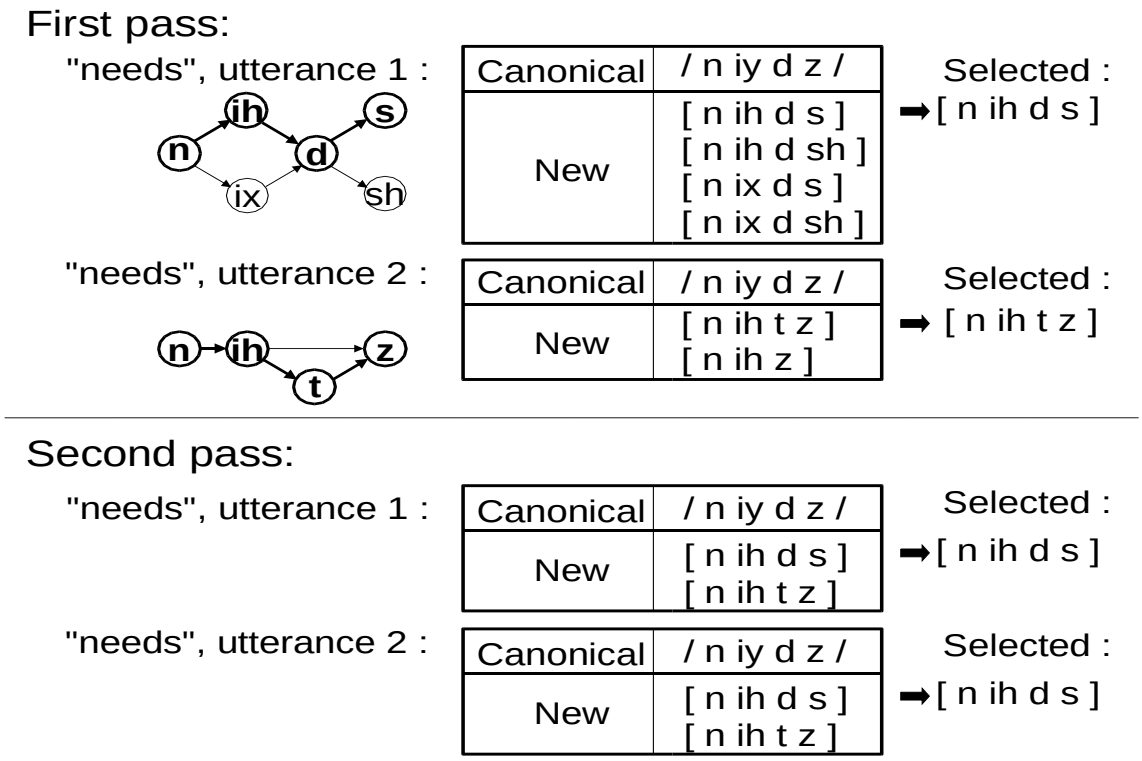

Figure 4.10: Selection of pronunciation variants using two passes of Viterbi alignment

tions were therefore subject to further pruning depending on their frequencies of occurrence. Namely, pronunciation variants whose probability of occurrence was lower than a minimum value P_{min} were rejected. Probabilities were estimated by simply dividing the number of times a phonetic transcription of a word was chosen by the number of word tokens in the training material. The final remaining pronunciation variants and all canonical transcriptions constituted the final static augmented lexicon.

4.6 Dynamic lexicon building

4.6.1 Overview

The objectives in this second part was to check *during recognition* whether a word was likely uttered, and if so to use only phonetic transcriptions of the word that best matched the pronunciation characteristics of the input utterance. For this purpose, the static augmented lexicon described in the previous subsection was accessed and filtered through the following steps and for each *testing* utterance (Figure 4.11):

1. A pronunciation network was created following the same method used to build the static augmented lexicon. This involved again the automatic detection of phonetic features from speech and the successive steps described in section 4.5 to generate the possible alternatives.

2. For each word in the static augmented lexicon, the network was scanned to find a match with one of the available phonetic transcriptions of the word. If the match was good enough, the best matching transcription according to some criteria was selected and

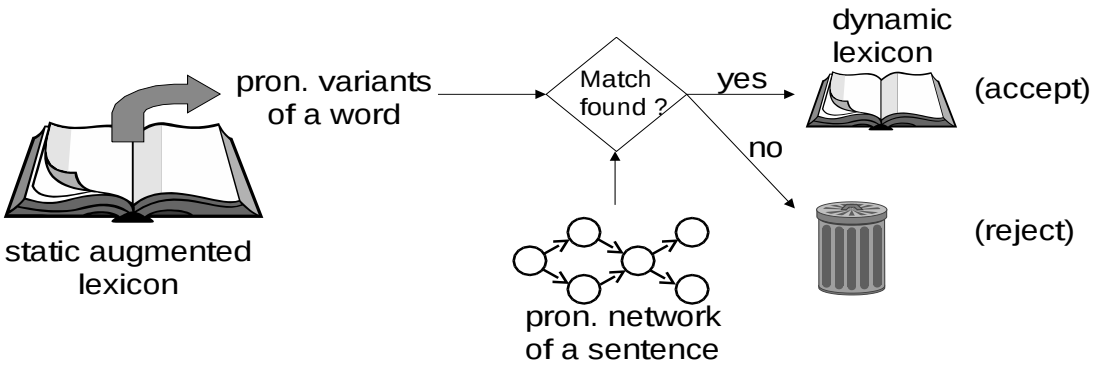

Figure 4.11: Steps to build a dynamic lexicon

added to the basic lexicon, otherwise only the canonical transcription was kept and all pronunciation variants of the word were rejected.

The resulting lexicon at the end of the process was used instead of the basic and static augmented lexicons in a standard HMM recognition. It was called *dynamic* because its content was adapted to each distinct utterance, that is, different entries were selected or eliminated depending on the utterance. The lexicon filtering idea follows a similar line of thought as in Kirchhoff [83] and Lahiri [95], but is different in its objective and method. First, their objective was to select word or syllable candidates in the lexicon while this method selected *pronunciation* candidates of words, but without necessarily eliminating any word (no word was actually rejected in our experiments). Second, they accessed the lexicon at the feature level while this method detected phonetic features from the input speech, but accessed the lexicon at the phone level (due to the practical reasons mentioned in section 4.5.3).

The next subsection will explain how to access and filter the static augmented lexicon to build dynamic lexicons.

4.6.2 Pronunciation match search

The static augmented lexicon was filtered by searching the phonetic transcription of each lexical entry in the pronunciation network. The search method described in this subsection was inspired from James and Young [72] who presented a wordspotting technique that searches phonetic transcriptions of words to spot in a phone-level lattice. Dynamic programming was used to perform the search based on likelihood scores associated with phone hypotheses of the lattice and some empirically defined penalties to account for phone insertions, deletions and substitutions. They reported that the search was much faster than more conventional wordspotting techniques with still reasonable performance. To further speed up the search process, the method adopted here was also inspired by the work of Dharanipragada and Roukos [34] who experimented a two-level match strategy: first, a coarse acoustic score located possible time intervals where the word to spot could have been uttered, then a more detailed acoustic match was performed at these intervals to reduce the number of false alarms. They reported a high detection rate given their experiment conditions (words to spot not used in training their acoustic models, no language model used) and especially a high speed of execution (much faster than real-time). The method presented here will however be a bit different from the above techniques in that phone-level and feature-level scores will be used instead of acoustic likelihoods (cf. section 4.6.3). The method is described by the following

steps for each lexical transcription (Figure 4.12):

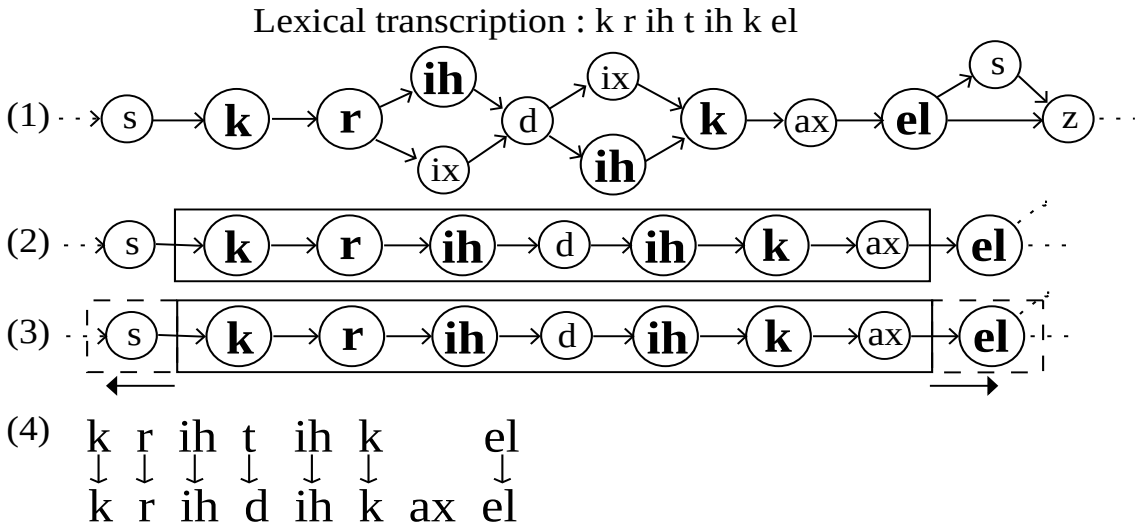

Figure 4.12: Steps to search for a lexical transcription in a pronunciation network

1. Nodes in the network representing any phone of the lexical transcription were marked, regardless of their relative positions.

2. The network was scanned to locate accumulation of marked phones. For this purpose, a window of size equal to the number of phones of the lexical transcription was set around each marked phone in the network[4]. A coarse matching ratio was calculated by dividing the number of marked phones by the total number of phones in the window (if a phone was marked several times in the window, it was not counted more than the number of times it appeared in the lexical transcription). If the resulting ratio was equal or above a certain threshold MR_{min} ("MR" stands for "Matching Ratio"), it was considered as a putative hit and the window was kept for further process, otherwise the window was rejected. If no window satisfied this criterion, the lexical transcription was rejected.

3. Each window accepted in the previous step was progressively extended on both sides to try to find other distinct marked phones in the area. The extensions continued as long as the corresponding coarse matching ratio increased (as mentioned in the previous step, a given marked phone was not counted more than the number of times it appeared in the lexical transcription) or a single phone insertion was detected. Only windows with the highest matching ratio were kept, the others were discarded.

4. A detailed match score was finally evaluated through dynamic programming (DP) between the lexical transcription and each sequence of phones found in the selected windows. Distances between phones were calculated by comparing their SPE features (number of different features divided by the total number of features), with the highest penalty when a consonant was mapped to a vowel. Similar matching ratio as in step 2 was calculated for each window (number of correctly matching phones / total number of phones), but was this time more precise because a correct ordering of phones was required by the DP alignment. If at least one window had a ratio above the threshold MR_{min} and the

[4]Position(s) of the window relative to the marked phone reflected the relative position(s) of the phone in lexical transcription. A window position was never checked more than once for a given lexical transcription and pronunciation network.

corresponding alignment was realistic (*e.g.*, no two consecutive insertions), the match was considered as valid.

An example is given in Figure 4.12 for a transcription of the word "critical". After marking all phones of the lexical transcription [k r ih t ih k el] in the pronunciation network (step 1), a valid window was located with a coarse matching ratio of 5/7 (step 2). After the window was extended (step 3), the ratio increased to 6/8. The final DP match (step 4) confirmed this ratio, although it could have been different if the phones in the network were not ordered in the same way as in the lexical transcription.

4.6.3 Detailed-level matching scores

The matching ratio previously described was too approximate to accurately identify the most appropriate pronunciation of a word given an utterance, so it was frequent that several phonetic transcriptions of the same word found a match in the pronunciation network. In order to keep only the most suitable pronunciation per word, two more detailed matching scores were used. The first one, S_{phn}, was evaluated at the phone level. It concerned only phones that were correctly mapped according to the DP alignment mentioned in the previous subsection (*i.e.*, substitutions, deletions and insertions were ignored). It was evaluated from the conditional probabilities of appearance of these phones given the word W :

$$S_{phn} = \sum_{i \in \Gamma} \sum_{j=1}^{N_{pron}} P(ph_i|pr_j) \cdot P(pr_j|W) \tag{4.1}$$

Γ is the set of all lexical phones that were correctly matched, N_{pron} is the number of distinct pronunciations for the word W, ph_i is the i-th lexical phone correctly matched and pr_j is the j-th pronunciation of the word W. The probabilities were estimated by evaluating frequencies of occurrence of phones and pronunciations for each word during the generation and selection of new pronunciation variants in section 4.5.

Although this phone-level score was sufficient to select the best matching transcription in many cases, there were still some situations where two candidate pronunciations led to the same score. In this case, a second score, S_{feat}, was evaluated at the feature level: it consisted of comparing phonetic features of each pair of aligned phones and of estimating a global similarity measure. This new score was more precise than the phone-level score S_{phn} because it also took account of all mapping errors (*i.e.*, substitutions, deletions and insertions), but was on the other hand more computationally expensive (hence the two-level detailed scores were adopted). Let us first consider self-aligned phones and phone substitutions only. In case a phone A was mapped to a phone B (A could be identical to B), the *theoretical* feature vector (combination) of A (*i.e.*, as given by a phone-features conversion table) was compared to the sequence of feature vectors *detected* by the ANN frame-by-frame and associated with B. Since B spanned several time frames, the feature vector of A was copied to cover the same time interval. Assuming that feature vectors of A and B were respectively noted $\{\overrightarrow{a}, \overrightarrow{a}, ...\}$ (N times to cover N frames) and $\{\overrightarrow{b_1}, \overrightarrow{b_2}, ..., \overrightarrow{b_N}\}$, and moreover supposing that successive frames were independent, a *measure of similarity* (MoS) between A and B was defined as follows:

$$\log MoS(A, B) = \frac{1}{N} \sum_{n=1}^{N} \log MoS(\overrightarrow{a}, \overrightarrow{b_n}) \tag{4.2}$$

Assuming that phonetic features of a given frame were also independent and supposing there were M features per vector, the MoS of each pair of feature vectors $\overrightarrow{a}$ and $\overrightarrow{b_n}$ was obtained by the following expression:

$$MoS(\overrightarrow{a}, \overrightarrow{b_n}) = \frac{1}{M} \sum_{m=1}^{M} [1 - |targ(a^m) - act(b_n^m)|] \tag{4.3}$$

This MoS is simply based on the average of differences between features of both vectors. $targ(a^m)$ is the m-th target (theoretical) feature value of vector $\overrightarrow{a}$ (0 if the feature is absent, 1 if the feature is present), and $act(b_n^m)$ is the m-th activation value of vector $\overrightarrow{b_n}$, as returned by the ANN for the m-th feature (values originally ranged from -1 to +1, but were rescaled to fit in the interval [0, 1]). The final feature level score, S_{feat}, was the average MoS over all phone pairs given by the DP alignment:

$$S_{feat} = \frac{1}{K} \sum_{k=1}^{K} \log MoS(A_k, B_k) \tag{4.4}$$

where K is the number of aligned phone pairs. The final expression for S_{feat} can be found by including the expressions of equations (4.2) and (4.3) in (4.4).

Phone insertions were handled like substitutions by including the notion of *transitional phone* introduced in section 4.5.3. We recall that a transitional phone represented any combination of phonetic features that could not be matched to any phone in the phone-features conversion table, and was mainly due to asynchronism of articulatory movements. A possible adequate phone-level representation of such movements at phone boundaries could be a sequence comprised of two valid phones A_1 and A_2 in between which one or more transitional phones could be inserted to represent the sequence of transitional articulatory configurations; several parallel sequences of transitional phones would even be necessary to model the movement of an articulator before another and vice-versa. However, to simplify the problem and for the purpose of modeling phone insertions in the same way as substitutions, this hypothetically complex representation was simplified by assuming only one single transitional phone T between the phones A_1 and A_2, with the following target feature values:

$$targ(t^m) = \begin{cases} 0 & \text{if } targ(a_1^m) = targ(a_2^m) = 0 \\ 1 & \text{if } targ(a_1^m) = targ(a_2^m) = 1 \\ 0.5 & \text{otherwise} \end{cases} \tag{4.5}$$

This expression simply means that the transitional phone takes the same m-th target feature value as its surrounding valid phones if their m-th features are identical, or takes an intermediate value (0.5) if they are different. This is in accordance with the results of King and Taylor [82], who observed that their network output feature values tended to be intermediate at phone boundaries. In the context of our lexical transcription match in a pronunciation network and resulting DP alignment with the best match (as shown in Figure 4.12), insertion of a phone B (from the best network match) between two lexical phones A_1 and A_2 was evaluated by measuring the resemblance of B to the target transitional phone T:

$$\log MoS(\emptyset, B) \approx \log MoS(T, B) = \frac{1}{M}\frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N} [1 - |targ(t^m) - act(b_n^m)|] \tag{4.6}$$

where M and N are respectively the number of features and the number of feature vectors (frames) compared, just like in equations (4.2) and (4.3). The idea behind this equation is to see how much the detected feature values of phone B are distinct from typical feature values of a transitional phone. If comparisons show little differences, B can be assimilated to a transitional phone that could be present between the lexical phones A_1 and A_2. The insertion of B is therefore plausible and probability of insertion is high. On the contrary, if B is significantly different from a transitional phone, it is likely a valid phone and the fact that B is not present in the lexical transcription must be penalized, hence a low insertion probability expressed by a low MoS. An example is shown in Figure 4.13 that follows the case given in Figure 4.12. Results of DP alignment showed an insertion of the phone [ax] between two lexical phones [k] and [el]. To evaluate how much penalty the lexical transcription should be accounted for this insertion, an idealistic transitional phone [k_el] is introduced between [k] and [el], with phonetic features set according to the expression (4.5) and given in the figure. Insertion of a phone [ax] is then made equivalent to a substitution of [k_el] into [ax] and the MoS for this substitution is evaluated by comparing their phonetic features. If the resulting MoS is high, features associated with [ax] represent well a transitional phone that naturally stands between [k] and [el]; in other words, the insertion is not considered as a real valid phone insertion and hence the penalty is low. On the other hand, if the MoS is low, detected features likely represent a valid distinct phone [ax] (or similar phone) and lack of such phone in the lexical transcription is therefore penalized.

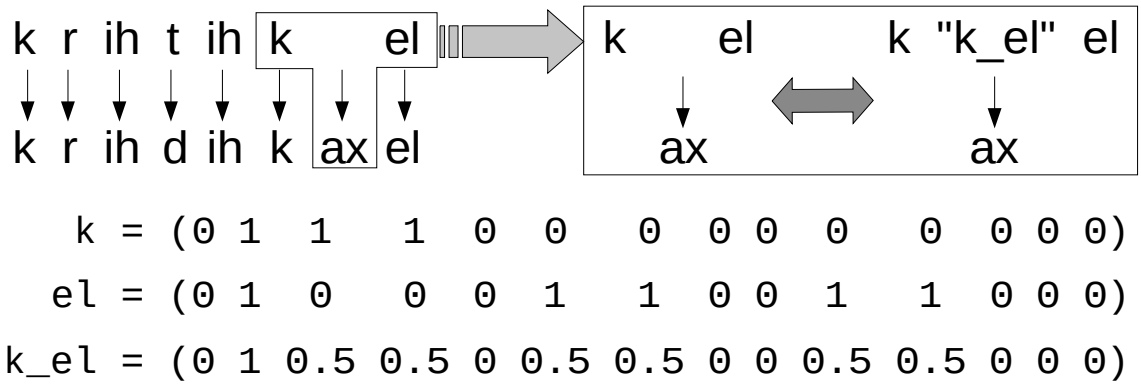

Figure 4.13: Procedure to evaluate measures of similarity for phone insertions using the concept of transitional phone

Phone deletions were handled similarly: deletion of a lexical phone A was interpreted as a substitution of A into a transitional phone T standing between two network phones B_1 and B_2:

$$\log MoS(A,\emptyset) \approx \log MoS(A,T) = \frac{1}{M}\frac{1}{N}\sum_{m=1}^{M}\sum_{n=1}^{N}[1-|targ(a^m)-act(t_n^m)|] \tag{4.7}$$

In contrast with insertions, phonetic features for T were not artificially created, but directly taken from the outputs of the ANN. Namely, features of all frames located between the end of phone B_1 and start of B_2 were used (start and end times of two successive phones did not always coincide, see Figure 4.7 for an example). Alternatively, some boundary frames of these phones were used instead if no such frames existed. These frames were not necessarily transitional ("transitional" as defined in section 4.5.3) because they could be associated with a valid phone hypothesis distinct from B_1 and B_2 in the pronunciation network. The MoS

between the lexical phone A and T determined therefore if features associated with T represented a true transitional phase between B_1 and B_2 - in which case presence of the phone A in the lexical transcription was penalized by a low MoS - or rather resembled to feature values of the phone A or phonetically similar phone - in which case a low penalty by means of a high MoS was applied.

The following sections will describe some experiments and results relative to the methodology presented.

4.7 Experiments and basic results

4.7.1 The TIMIT database, lexicon and phone inventory

All experiments were carried out on TIMIT [51]. This is a read speech database spoken by 630 speakers from eight major dialect regions of the United States. Each speaker uttered ten sentences of three types:

1. Two "SA" sentences were uttered by all speakers and are convenient to compare pronunciation variations between the different dialects.
2. Five "SX" sentences are phonetically compact: they are meant to insure a good range of phone pairs, but without exhaustively covering all phonetic contexts either. Each sentence was uttered by seven different speakers.
3. Three "SI" sentences are phonetically diverse: they were selected from existing text sources and are meant to cover a maximum range of phonetic contexts. Each sentence was uttered by a unique speaker.

The database is divided in two sets, train and test, and contains 462 and 168 speakers respectively. The proposed *complete* test set contains 1344 SX and SI sentences. A smaller *core* test subset was created with 24 speakers (two male and one female speakers of each dialect), with all SX and SI sentences per speaker. The database contains reference transcriptions and segmentations at both word and phone levels, hence convenient for evaluation purposes.

The lexicon provided with TIMIT was used for the experiments. It contains a closed vocabulary of 6229 words. Each word is associated with one baseform pronunciation in most cases. 62 phones (including silence and pause) composed the total phone inventory (cf. appendix A), although it is common for training and evaluation purposes that this original set is reduced (*e.g.*, 48 symbols proposed by Lee and Hon [97]). However, a different 43 symbol set was adopted in this chapter, in accordance with the SPE phone-features conversion table given by King and Taylor [82], in which several phones were mapped to the same feature combination. In order to establish a one-to-one relationship between feature combinations and phones, all phones with the same feature combination were merged to build a single acoustic model. The list of phones and feature combinations used can be found in appendix B.1.

4.7.2 The baseline system

The HTK speech recognition system [158], well-known and often used in the speech community, was used to build HMM-based acoustic models. The original speech data was first converted

into acoustic vectors using a Hamming window of 25ms and a 10ms frame interval. Each vector contained 39 elements: 13 Mel-frequency cepstral coefficients (MFCC) including normalized energy, plus the corresponding first and second derivatives. The 0-th order cepstral coefficient (c_0) was not included.

Influenced by the tendency of using TIMIT by the speech community for phone recognition, we decided to rely on a training scheme proposed by Young and Woodland [160] as a guide to build our acoustic models, then to evaluate them in a word recognition task. Acoustic parameters and reference phonetic transcriptions of all training sentences of TIMIT were used to first build 43 monophones, then based on them to create and train 1304 right-context biphones as well. All HMMs had the standard left-to-right topology with three states and no skips. A data-driven clustering scheme was then applied to merge acoustically similar states, then the number of Gaussian mixtures per state was progressively increased up to six. A back-off bigram was built from all distinct sentences of TIMIT for evaluation[5]. The baseline results are given in Table 4.4. The phone accuracy of 71.9% (28.1% PER) obtained with the full test set of TIMIT is comparable to the result reported by Young and Woodland. The same acoustic models evaluated for word recognition led to 25.1% of WER.

	PER / WER
Phone recognition	28.1%
Word recognition	25.1%

Table 4.4: Baseline recognition results

4.7.3 The phonetic feature detection system

Similar experiments as in King and Taylor [82] were carried out to perform the feature detection. As mentioned earlier, an ANN (called NICO [107]) was used to map a set of acoustic parameters to a vector of phonetic features on a frame-by-frame basis. The ANN created for the experiments had the three standard layers: input, hidden and output. The input layer was composed of 13 units to accept 12 MFCCs and normalized log energy (although more input parameters were used in total, see below), the output layer had 14 units, one for each SPE feature and a special unit for silence, and the hidden layer had 250 units. The tanhyp function was used as the activation function for all these units. Besides this basic configuration, the ANN architecture included the following characteristics:

- Two special additional layers with the same number of units as the input layer were created. Each unit of the first additional layer was connected to a distinct unit of the input layer with time-delay and look-ahead connections in such a way that the activity of the new unit represented the first derivative of the corresponding input unit. The second additional layer was similarly connected to the first additional layer to calculate the second derivatives. All units of the input and additional layers were connected to the units of the hidden layer.

- Several time-delay and look-ahead links connected pairs of units in successive layers to take account of context frames. A window of -3 to +3 frames (so seven links in total) fully connected all pairs of units in successive layers.

[5]Test sentences were voluntarily included so that the out-of-vocabulary problem would not influence the results of our experiments.

- Units in the hidden layer were also connected between each other (recurrent links), but with only 50% connectivity. Units were ordered and probability of connection between each pair was proportional to the proximity of units.
- The network contained about 150'000 links in total.

To train the ANN, reference phonetic labels and segmentations of all speakers found in the TIMIT training database were used to set the target feature values according to the phone-features conversion table they provided (0 if a feature was absent, 1 if it was present), given the acoustic parameters generated with HTK and presented to the input layer at each frame. 3596 sentences were used to train the network and 100 sentences were reserved for cross-validation. Even though target features were set to change synchronously at phone boundaries according to the given segmentations, the network was still able to learn the asynchronous change behavior of phonetic features during recognition.

4.7.4 Phonetic feature recognition results

Frame-level phonetic feature classification experiment on the core test set of TIMIT led to the results in Table 4.5. Values are comparable to those obtained by King and Taylor on their cross-validation set, and show that each feature taken separately can be reliably recognized. The "all correct" rate shows how frequently all features were *simultaneously* correct for a given frame ("all correct" = number of frames with all features correct / total number of frames); about one frame out of two was phonetically well-identified.

Feature	Frames correct (%)	Feature	Frames correct (%)
vocalic	88.2	round	93.9
consonantal	90.5	tense	90.7
high	88.0	voice	93.6
back	87.9	continuant	93.3
low	93.4	nasal	97.7
anterior	90.6	strident	97.0
coronal	89.9	silence	98.3
Average	**92.4**	**All correct**	**53.5**

Table 4.5: Frame-level classification results with SPE features

4.7.5 Word recognition results

The static augmented and dynamic lexicons generated following the methods described in sections 4.5 and 4.6 were used instead of the basic lexicon for word recognition. The following parameter values were chosen for the experiments (values were not optimized for the experiments, but chosen either arbitrarily or guided by common sense):

- Limits of the incertitude zone to generate multiple feature combinations (section 4.5.2): $T_{min} = -0.5$ and $T_{max} = +0.5$
- Minimum number of successive frames associated with a same phone to create a valid hypothesis (section 4.5.4): $F_{min} = 3$

- Minimum number of frames for a reliable hypothesis (section 4.5.4): $R_{min} = 2$
- Pruning probability for frequent words (section 4.5.6): $P_{min} = 0.05$

The static augmented lexicon contained on average 2.4 pronunciations per word; it included all canonical transcriptions and the pronunciation variants selected by two passes of Viterbi alignment (cf. section 4.5.6). For dynamic lexicons, two cases were evaluated. In the first case, only one pronunciation per word was accepted, regardless of whether the pronunciation was canonical or not. In the second case, canonical transcriptions were mandatory and a single pronunciation variant was added only if it found a better match in pronunciation networks than the corresponding baseform. Word error rates using these lexicons on the full test set of TIMIT are given in Table 4.6. The results show that the static augmented lexicon (line "Static + canonical") improved performance but not by much. The dynamic lexicons (line "Dynamic") were built using different values of matching ratio thresholds MR_{min} (introduced in section 4.6.2). These lexicons alone did not perform better than the baseline system (27.7% WER). However, combination of dynamic lexicons and canonical transcriptions (line "Dynamic + canonical") achieved a 21.7% WER, so a 13.5% relative reduction in WER compared to the baseline and 10.0% compared to the static augmented lexicon (statistically significant improvements). The corresponding MR_{min} was around 30% for our system.

Lexicon	WER
Basic (canonical)	25.1
Static + canonical	24.1
Dynamic	27.7
Dynamic + canonical	21.7

Table 4.6: Recognition results with static and dynamic lexicons

4.7.6 Expected maximum performance

To have an idea about the best performance we could get from the concept of dynamic lexicons using canonical and derived transcriptions, the following cheating word recognition experiment was set up. Like in the basic experiment with dynamic lexicons, we used a different lexicon for each test utterance during recognition. Lexical entries were identical to those in the basic lexicon, except for the words pronounced in the utterance: if a phonetic transcription of a word, derived from the pronunciation network built for the utterance, better matched through DP alignment the reference phonetic transcription than the baseform, then the latter was replaced by this pronunciation variant in the lexicon. Application of this method led to the results in Table 4.7 (lines "Basic (canonical)" and "Pron. network"). We notice that a significant improvement can potentially be achieved (more than 50% relative reduction in WER). For the sake of information, the same experiment was carried out again, but this time using directly the reference phonetic transcriptions instead of network-derived pronunciations. As expected, relative improvements are even higher (line "Reference"). It is interesting to note that when reference transcriptions are *added* to existing canonical transcriptions instead of *replacing* them (line "Reference + canonical"), further improvement can be obtained (13.8% relative reduction in WER compared to reference transcriptions alone), suggesting that it is a good idea to always keep canonical pronunciations even when lexicons are dynamic and pronunciation variants are reliable.

Transcriptions	WER
Basic (canonical)	25.1
Pron. network	12.5
Reference	5.8
Reference + canonical	5.0

Table 4.7: Expected maximum word recognition performance with canonical, pronunciation network-derived and reference transcriptions

4.8 Analysis of intermediate results and errors

Comparison of results in Tables 4.6 and 4.7 shows that there is still a lot of room for further improvement. The objective of this section is to analyze some intermediate results and errors to identify the components of the current system that could be improved.

4.8.1 Detection of phonetic features and comparison with phones

In order to see whether use of phonetic features is more useful and efficient than phones as often stated in literature, a separate phone-based ANN was also trained. It was similar to the feature-based system described in section 4.7.3, except that 43 units (one per phone) composed the output layer instead of the previous 14 SPE units. Pairs of units in successive layers were again fully connected, resulting in more links (200'000 instead of 150'000) due to the higher number of output units. The training process respected a standard N-to-1 scheme (in contrast with the N-to-M scheme used with SPE features), that is, only one output unit was activated at each frame. Classification results on the core test set of TIMIT showed a 67.6% accuracy, so much higher than the 53.5% obtained with the SPE-based system. However, this difference in performance was partially due to the number of possible outputs : while there were merely 43 possibilities for the phone-based system at each frame, $2^{14} = 16384$ different SPE-feature combinations were possible with only one correct for the feature-based system, hence a much higher risk to make an error. This was especially the case at phone boundaries due to a higher rate of transitional frames. King and Taylor [82] reported that forcing each feature combination to be mapped to the closest phone of their inventory increased their frame recognition rate from 52% to 59%. Moreover, phonetic features used as references during recognition were simply deduced from reference phone-level information using a phone-features conversion table. These reference feature values were not always correct. First, their configurations were set to change *synchronously* at given phone segmentation points while they were supposed to change *asynchronously* in reality. Second, since the evaluated speakers were from various dialectal regions of the United States, substantial pronunciation variations were expected and some phonemes could have been realized with particular articulatory configurations not found in the phone-features conversion table. Real measured articulatory positions would have given more accurate results. Nevertheless, the substantial difference in frame recognition rates obtained between the feature- and phone-based ANNs still suggests further study to improve the feature detection system.

Given the higher accuracy achieved with a phone-based ANN, one could rightfully ask why a feature-based ANN could be more beneficial. If the final goal is just to get the most accurate phonetic identification of frames, then a feature-based ANN is probably not useful according to the results above. However, our objective was to create pronunciation networks from the outputs of an ANN. Such networks imply alternative paths to model pronunciation variations.

Consequently and in this specific context, it is not important if the correct phone is not ranked first for a given frame during ANN recognition, as long as it is among the N best alternatives. On the other hand, it is more important that the N-best candidates are phonetically similar to create acceptable pronunciation alternatives in networks. The following experiment was therefore set up to measure the average phonetic confusion between alternatives: two phone confusion matrixes were built (silence symbol excluded) by using both the phone- and feature-based ANNs for recognition. However, in contrast with the standard way, not only the best output but the N-best phone-level alternatives per frame were taken into account to build each matrix. Then, a global average confusion distance was calculated for each matrix by respecting the following steps:

1. For each matrix row (elements of a row corresponded to the list of confusable phones p_i's given a reference phone p_{ref}), each confusion count, $CCount(p_{ref}, p_i)$, was normalized by the total number of confusion counts found in the row to get a confusion ratio: $CRatio(p_{ref}, p_i) = \frac{CCount(p_{ref}, p_i)}{\sum_{i=1}^{N} CCount(p_{ref}, p_i)}$, where N is the number of distinct phones.

2. An average confusion distance for each reference phone, $CDist(p_{ref})$, was calculated by summing the products of each confusion ratio $CRatio(p_{ref}, p_i)$ by the corresponding number of phonetic feature (SPE) differences between the reference and confusable phone, $FDiffs(p_{ref}, p_i)$: $CDist(p_{ref}) = \sum_i CRatio(p_{ref}, p_i) * FDiffs(p_{ref}, p_i)$. To clearly mark the differences between consonants and vowels, the original SPE binary features given by King and Taylor [82] were made ternary: all features theoretically not relevant for a consonant or a vowel were marked as '0' for the concerned phones regardless of their original values ('+' or '-'), unless the modification reduced too much discrimination between consonants or between vowels themselves (phones and their ternary features are listed in appendix B.1). These non-relevant features were therefore counted as additional differences when a consonant was confused with a vowel or vice-versa.

3. A global confusion distance was finally calculated by taking the average over all confusion distances calculated for each reference phone: $CDistGlob = \frac{1}{N} \sum_{j=1}^{N} CDist(p_{ref,j})$, where N is the number of reference phones (N is the same as in point 1).

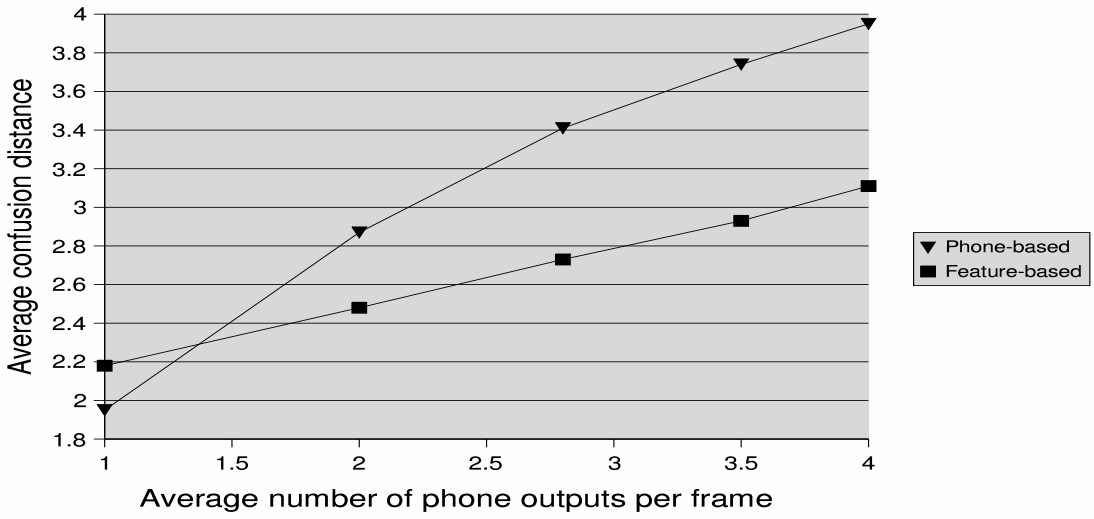

Figure 4.14: Average confusion distance with respect to the average number of phone outputs per frame, using a phone-based vs. a feature-based ANN

Results are given in Figure 4.14 for different average numbers of phone outputs per frame. Partially due to its better frame recognition accuracy, the phone-based ANN shows a smaller confusion distance than the feature-based ANN when only the best output per frame is considered. However, this distance increases also more rapidly when alternatives are also taken into account, which means for instance that the second or third best outputs are phonetically more distant on average to the reference phone when using a phone-based ANN. Choice of the type of ANN to use should therefore depend on the objective aimed; according to the results, feature-based ANNs seem more suitable to generate phonetically closer alternatives.

4.8.2 Accuracy of pronunciation networks

Pronunciation accuracy of networks built from phonetic features was measured. It was necessary that transcriptions derived from these networks better modeled true pronunciations than baseforms in order to expect good pronunciation variants in static augmented and dynamic lexicons and hopefully an increase in speech recognition performance. For this purpose, a representative portion of each word subnetwork was compared against its corresponding reference phonetic transcription. Namely, ten paths with the lowest feature-level penalty scores (penalty scores were explained in section 4.5.2) were selected in each generated subnetwork; the ten phonetic transcriptions were aligned against the reference transcription using DP and the mapping errors (deletions, insertions and substitutions) were counted from the best alignment. As a matter of comparison, the canonical transcription was separately aligned against the reference transcription for the same purpose. The resulting Phone Error Rates (PERs) with the core test set of TIMIT are shown in Table 4.8. Pronunciation networks alone did not on average target true pronunciations better than canonical pronunciations, suggesting that more effort needs to be put on how to create and combine phone segment hypotheses more efficiently to build the networks. However, another experiment that combined both canonical and network-derived pronunciation variants and aligned them against reference transcriptions led to statistically significant improvement over the use of canonical transcriptions alone (28.8% relative reduction in PER). Pronunciation networks generated from phonetic features were therefore still useful but needed at this stage to be combined with canonical transcriptions to bring higher pronunciation accuracy.

Transcriptions	PER
Canonical	29.05
Pron. network	38.09
Can. + pron. network	20.67

Table 4.8: Phone error rates with canonical and pronunciation network-derived transcriptions

4.8.3 Accuracy of lexicons

Pronunciation accuracy of dynamic lexicons was measured. In contrast to the case with pronunciation networks seen in the previous subsection, possible phonetic transcriptions were restricted to the lexical entries available in the static augmented lexicon (which was built using the training set). Performance was therefore expected to be lower than with pronunciation networks if transcriptions did not model well the pronunciation variations of the test set. Pronunciation accuracy was measured in a way similar to the previous subsection: for each reference word of the test set, the phonetic transcription of the word in the dynamic lexicon that best matched its corresponding true phonetic transcription through DP alignment was

selected and the corresponding mapping errors were counted. In order to estimate the maximum achievable performance using all pronunciation variants generated from the training set, the same experiment was set up but this time with the static augmented lexicon (which contained all the available transcriptions).

Transcriptions	PER
Can. + pron. network	20.67
Static aug. lexicon	22.06
Dynamic lexicons	26.10

Table 4.9: Comparisons of phone error rates with network-derived pronunciations and with transcriptions found in static augmented and dynamic lexicons

Corresponding PERs obtained with the core test set of TIMIT are shown in Table 4.9 and are compared to the PERs obtained with canonical and network-derived transcriptions. The table shows that a slight degradation is observed with the static augmented lexicon compared to the network-derived transcriptions. This is due to the mismatch between transcriptions built from the training set and pronunciations of the test set. Besides, this result represents the case when the pronunciation search algorithm (explained in section 4.6.2) always selects the phonetic transcriptions that best match the true pronunciations of the test set. This algorithm was not without errors in practice: transcriptions selected for dynamic lexicons led to a higher degradation in PER compared to the network-derived pronunciations. Therefore, both a better generalization of pronunciation modeling to unseen data and more accurate pronunciation search algorithm are required to reduce this degradation.

4.8.4 Accuracy of pronunciation search algorithm

Following the results of the previous subsection, we measured how frequently the pronunciation search algorithm selected the best lexical transcription given the true pronunciation of a word: experiments showed a 63.1% of correct identification rate. Errors in the remaining percentages were of two types:

False alarm : the canonical transcription best matched the true pronunciation, but the algorithm selected a pronunciation variant as the best match: 23.4%.

Bad variant match : a pronunciation variant best matched the true pronunciation, but the algorithm selected either the canonical transcription or another pronunciation variant instead: 13.5%.

These two rates were partially influenced by the pronunciation match threshold MR_{min} (mentioned in section 4.6.2): the lower this threshold and the more easily a pronunciation variant was accepted and the false alarm rate was therefore increased. On the other hand, a high threshold rejected most pronunciation variants among which some of them could better model the true pronunciation than the canonical transcription, which increased the bad variant match rate. Besides, these values were average rates. Efficiency of the pronunciation match depended actually a lot on the word length. Figure 4.15 shows how the three rates mentioned above (correct, false alarm, bad variant match) varied with respect to the word length (number of letters): the longer the word and the more accurate was the match. As it could have been expected, short words were more difficult to spot because their phonetic transcriptions were often located at several places in pronunciation networks among which most of them were misleading (*e.g.*, longer words could include similar phonetic patterns).

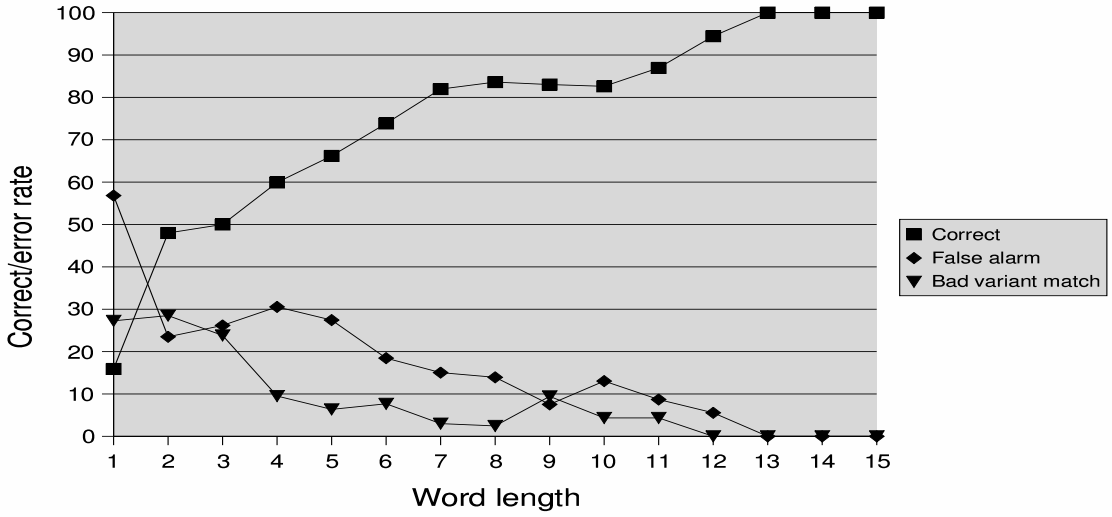

Figure 4.15: Variation of correct, false alarm and bad variant match rates with respect to word length

4.9 Discussion and possible future directions

To summarize and according to the result analysis made, the following points need to be improved to expect a better pronunciation modeling accuracy and word recognition rate:

- More reliable detection of phonetic features
- More accurate pronunciation networks
- Better generalization of dynamic lexicons to unseen data
- Better location of short word transcriptions in pronunciation networks

Let us consider each item separately. Concerning the feature detection system, it is known from the literature (*e.g.*, [46]) that training with real articulatory data lead to higher performance than with phone-derived simulated feature data. But needless to say, such data is not always available for the database we would like to use for evaluation. Perhaps that first bootstrapping an ANN with real articulatory data of another database (preferably of similar type) before further training with simulated articulatory data of the speech corpus we are interested in could help to improve the detection. Also in the current system, all feature targets had binary values and moved synchronously and instantaneously from one state to the other at phone segmentation points. A better way to simulate smoothly changing articulatory positions would be to use intermediate feature values (between 0 and 1) as targets in the vicinity of phone boundaries. Asynchronous movements could also be taken into account by setting each feature modification at different time intervals, but would require some a priori knowledge or statistical models to predict which feature is modified first given the contexts. Finally, to simplify the problem, all features were given the same importance regardless of the phone considered. It is known however that some articulators are *critical* to realize a phone (*e.g.*, tongue tip for [d] or [t], lips for [b] or [p]) while others are often more variable as a consequence or anticipation of realizing the previous/next phone. It was reported in [110] that "critical articulators are less variable in their movements than non-critical articulators". A weighting scheme could therefore be applied to give more importance on critical phonetic features and to help enhance the identification of some specific phones.

Concerning the creation of pronunciation networks, our current system simply linked phone hypotheses based on their relative positions in time, but without considering if a sequence of phones was well-formed or not. Application of a phone-level language model was expected to help in this context, but experiments with a phone-level bigram trained on all sentences of TIMIT did not bring any significant improvement in PER. A language model with higher order would perhaps bring some beneficial effects. A more appealing idea would be the application of *phonotactic constraints*, which are knowledge-based lists or rules that govern the set of permissible phone sequences in a given language. For example, consonant clusters of an initial "CCCV" (C=consonant, V=vowel) word or syllable sequence in English must start with a [s] followed by a voiceless stop ([p], [t] or [k]), then finally by one of [l], [r], [j] or [w] (with some additional constraints). A list of phonotactic constraints for English can be found in [30]. An example of approach with phonotactic constraints for speech recognition is given by Carson-Berndsen and Walsh [19] with their so-called Time Map model. They represented the set of phonotactic constraints as a finite-state automaton. An interesting aspect of their approach is the use of co-occurrences of phonetic features (two by two) as constraints on the arcs of the automaton. A data-oriented ranking procedure was used to relax some of the constraints in case no given phonetic sequence could respect all of them. However, in order to apply phonotactic constraints to continuous speech in English (and other languages), a syllable-level segmentation (*e.g.*, [40]) of recognized phone sequences is a priori necessary since phonotactic constraints in English are syllable-dependent.

Regarding the content of dynamic lexicons, a formalization procedure (using for instance rules or decision trees) would be more appropriate than addition of new phonetic transcriptions to the lexicon in order to generalize pronunciation variation modeling to words that did not occur in the training database. Moreover, since pronunciation networks built during recognition are phonetically more accurate than phonetic transcriptions generated during the training phase, entries of dynamic lexicons could be a mixture of the best phonetic entry found in the static augmented lexicon given an utterance and its best matching phonetic sequence in the pronunciation network. Transformation of the original phonetic entry could for example depend on the reliability of a phone segment hypothesis in the pronunciation network or more generally on some additional confidence metrics.

Better location of short word transcriptions in pronunciation networks is an open question. A possible approach (*e.g.*, Dharanipragada and Roukos [34]) is to build a network composed of an acoustic model for the keyword transcription and surrounded by "filler" models, to apply a Viterbi alignment in order to find the segmentation points and finally to rank the possible hits by time normalized log-likelihood scores. Another possibility could be to artificially extend the length of the initial phonetic transcription by adding one or more phones of possible neighbor words on both left and right sides; this idea is similar to the concept of multi-words (*e.g.*, Kessens et al. [80]), which are two or more words put together and considered as another distinct word on its own. Phonetic transcriptions would be longer and would facilitate the pronunciation search process and reduce the number of false alarms, but with the need of testing each possible neighbor context.

A last issue not yet mentioned is the speed factor. Although pronunciation search of a single transcription was faster than real time, construction of a dynamic lexicon took a considerable amount of time due to the high number of lexical entries to search for. At this stage, the method is therefore not suitable for real-time large vocabulary speech recognition. It is true however that current implementation of the method simply considered each lexical entry independently, one by one. Since many entries have similar phonetic sequences, a more efficient implementation (*e.g.*, based on a tree-organized lexicon [106]) would considerably

reduce the computation time. A recent paper by Koval et al. [92] also proposed a multi-level hierarchical representation of lexical entries that contained not only allophones and phonemes but also meta-phonemes at the highest levels of the hierarchy. Each node of the hierarchy was associated with a binary digit string of phonetic features (string length was proportional to the specificity of the node in the hierarchy) and a so-called hierarchical matching function that measured the importance of keeping or rejecting a feature attributed to a phoneme of the word. Such hierarchical representation could be appropriate to find a compromise between speed of execution and accuracy of the pronunciation search.

4.10 Summary

In this chapter, we introduced a method to build a lexicon whose content was adapted for each input speech utterance in order to better model pronunciation variations without increasing too much lexical confusability. For this purpose, two steps were applied. In the first step, a static augmented lexicon was created by adding new phonetic transcriptions to a basic lexicon. These pronunciation variants were generated using phonetic features that were automatically detected from speech and helped to create a pronunciation network. This process was followed by two Viterbi alignment passes on the network to select the best variants. Then in the second step, a distinct dynamic lexicon was created for each utterance during recognition by only keeping entries in the static augmented lexicon that best matched the pronunciation characteristics of the utterance. Decision of keeping or rejecting an entry was governed by a search of its phonetic transcription in a pronunciation network built from each utterance and based again on the detection of phonetic features. Although pronunciation networks alone did not target true pronunciations as accurately as canonical transcriptions, addition of the latter to the networks led to significant improvement of phone-level accuracy. Similarly, the system based on dynamic lexicons alone did not perform better than the baseline system, but addition of canonical transcriptions also led to significant reduction in WER. Analysis of intermediate results and errors pointed out which components of the system could be improved to expect a higher performance in the future.

Chapter 5

Dynamic Sharings of Gaussian Densities Using Phonetic Features

In the previous chapter, we described a method to dynamically model pronunciation variations using phonetic features at the *lexicon* level. This chapter will study the applicability of a similar approach, but this time at the *acoustic* level [101], through the following sections:

- Section 5.1 will explain the reasons that motivated this acoustic-level study.
- Section 5.2 will present the concept of state-level pronunciation modeling (SLPM).
- Sections 5.3 and 5.4 will describe two approaches that implement the SLPM concept, a static and dynamic approach respectively.
- Section 5.5 will describe the related experiments and some basic results.
- Section 5.6 will focus on modeling phone deletions and insertions in the dynamic SLPM framework.
- Section 5.7 is an independent part of this chapter dedicated to the detection of phonetic features in spontaneous speech.
- Section 5.8 will summarize the content of this chapter.

5.1 Motivations

Apart from the different points mentioned at the end of the last chapter (section 4.9), another issue of the previous method that seemed important to be addressed was the inability of acoustically modeling each phonetic feature configuration: they were constrained to be mapped to a finite number of available phone models, which represented a limited amount of possible articulatory configurations. Such limitation is an obstacle to correct pronunciation modeling according to an experiment made by Saraçlar and Khudanpur [123]: they showed that when a phoneme /b/ is realized as a phone [s], its average acoustics are neither close to the acoustics of a typical /b/ nor of a typical [s], but lie somewhere in between. Such instance is often a case of *pronunciation ambiguity*, for which even human transcribers do not agree about the identity of the surface form. Similarly in the domain of phonetic features, it is also likely that articulatory configuration during a pronunciation change is also an intermediate state between two "basic" configurations, and should therefore be modeled separately.

In the previous chapter, we already mentioned as a possible solution the work of Deng and Sun [33], who designed a specific state-transition network for each phone in order to model the overlap of articulatory features between neighbor phones. However, this approach requires in its basic form the definition of rules to decide when a feature can overlap a neighbor phone. Moreover, it is not designed to model pronunciation variations: it certainly does take account of pronunciation changes due to coarticulation effects, but not to other factors like dialects. For example, when a word "ten" is realized as [t ih n] instead of the basic form /t eh n/, the change from /eh/ to [ih] (or an intermediate form between /eh/ and [ih], as discussed above) is not due to the spread of articulatory features from its neighbor phones, but rather to the speech background of the speaker. Therefore, we were looking for a method more data-driven and more explicitly designed for pronunciation variation modeling.

With the perspective of addressing the issue mentioned in their previous paper, Saraçlar et al. [124] introduced a new method called *state-level pronunciation modeling* (SLPM). The next section will present its basic concept and section 5.3 will describe how it can be implemented. The novel contribution of this dissertation resides in the introduction of *dynamic SLPM*, which will be described in section 5.4.

5.2 Overview of state-level pronunciation modeling (SLPM)

5.2.1 Basic concept

The key idea of SLPM is to model pronunciation variations by sharing Gaussian densities across acoustic models. Namely, if a phoneme /b/ may be realized as a distinct phone [s], the phoneme shares the Gaussian densities of the phone to take account of this possible pronunciation change. The resulting output is a hybrid phone that inherits the acoustical properties of both the phoneme and phone. This is in contrast with the classical phone-level pronunciation modeling which supposes that only one of the original acoustic models may be used. The concept is illustrated in Figure 5.1 for the context-independent case, where the word "had" may be pronounced canonically (/hh ae d/), but also differently as [hh eh d]. Consequently, each state of the phoneme /ae/ may share the Gaussian densities of the corresponding state of the phone [eh]. When using context-dependent phones, the neighbor phones respect the same rule due to a change of their left and/or right phonetic context(s).

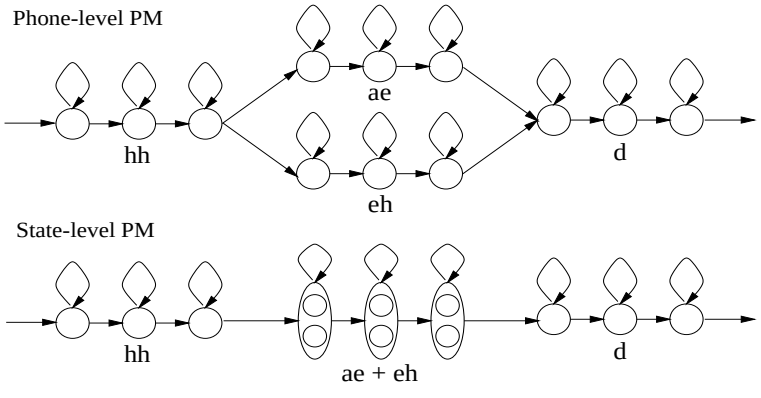

Figure 5.1: Phone- vs. state-level pronunciation modeling (from Saraçlar et al. [124])

5.2.2 Previous works on acoustic-level pronunciation modeling

Pronunciation modeling is often applied at the lexicon level, but is less common at the acoustic level. A standard approach to involve acoustic models in this context is to generate new phonetic transcriptions (hopefully closer to true pronunciations than canonical transcriptions) using a certain pronunciation modeling method, to retrain and refine the acoustic models with these new transcriptions, and to iteratively generate improved transcriptions using the acoustic models (*e.g.*, Sloboda [129]). However, this approach does not model pronunciation variations at the acoustic level as explicitly as SLPM.

Another type of method is to design acoustic models based on other topologies than the well-known three left-to-right HMM states. We already mentioned the work of Deng and Sun [33] who designed a specific topology for each context-dependent phone based on the concept of overlapping articulatory features. Eide [38] also created distinct context-dependent phone model topologies based on subphonemic units, each of which was modeled by a single state with a self loop. Creation of acoustic models followed a similar procedure as in Stolcke and Omohundro [134]: starting from a simple three state left-to-right configuration, a parallel path was added to the network each time that a subphonemic sequence associated with the considered context-dependent unit (associations were obtained from alignments of time segmented canonical phoneme sequences with the outputs of a phoneme recognizer) could not be explained by the network; states were then merged based on maximization of likelihood on a separate held-out data. Some improvement was obtained on both the Wall Street Journal and Broadcast News databases.

Several authors tried also other types of units than phones. Some researchers affirmed that syllables are better units to deal with pronunciation changes than phones. Greenberg [56] analyzed the Switchboard database and showed that syllables offer more systematic pronunciation variations than phones: a frequent case is when the onset is not modified, the nucleus is substituted and the coda is deleted[1]. A recent paper by Sethy et al. [127] applied a syllable-based approach for recognition of spoken names and found that it led to a much better recognition accuracy and speed than a phone-based system. Finke et al. [41] proposed another type of unit: their basic units were also phonemes, but augmented with other attributes (*e.g.*, articulatory features, stress) that were predicted using a trained pronunciation model. A corresponding acoustic model was determined using a decision tree technique and created for each augmented phone unit. This approach is close in its basic principle to SLPM in that a phoneme is not realized as another distinct phone, but only partial changes occur to the phoneme. Bacchiani and Ostendorf [7] designed customized acoustic units directly derived from data. First, an acoustic segmentation step divided all tokens of a given word into a fixed number of segments (this number varied with the word considered). Then, all segments of these tokens at the same i-th position were gathered to form an atomic (non-divisible) group of mean μ_p and covariance Σ_p. Next, all these groups were initially put in a global cluster that was iteratively divided in two by selecting the cluster with the lowest average likelihood per frame at each iteration. Finally, a K-means clustering algorithm was applied to remove clusters with too few observations. The final remaining clusters, retrained with the Viterbi or Baum-Welch algorithm, represented the final acoustic units and each word could directly be transcribed as a sequence of these units, thus allowing a joint design of acoustic unit inventory, acoustic models and lexicon. Their approach outperformed phone-based systems on a read speech database. No explicit pronunciation modeling was however taken into account since each word in the lexicon

[1] A syllable can be divided into three parts: the onset, the nucleus and the coda. The onset and coda generally correspond to a consonant and the nucleus to the vowel in between. For example in the word "cat" pronounced [k ae t], [k] is the onset, [ae] the nucleus and [t] the coda.

had a single pronunciation.

One of the reasons why explicit acoustic-level pronunciation modeling is less common is due to the existence of already efficient acoustic-level methods in the domain of speaker adaptation (*e.g.*, maximum likelihood linear regression (MLLR) [103]), which models to a certain extent intra-speaker pronunciation variations as well. But since speaker adaptation and lexicon-level pronunciation modeling may be complimentary, some works combined both types of methods to get better performance (*e.g.*, Humphries and Woodland [68], Willett et al. [150]). The method of Venkataramani and Byrne [141] is different from this approach because they explicitly used adaptation techniques for pronunciation modeling. Based on alignments between baseform and surface form sequences, they built a regression tree of MLLR transforms to predict acoustic changes associated with pronunciation variations. Each class of the tree represented a phonetic feature category based on vowel height and front-back positions. The phoneme-phone pair labeled on each link of their pronunciation lattices determined the tree node it corresponded to and the MLLR transformation found at the node was applied to the original phoneme acoustic model, instead of replacing the latter by the phone model in the pronunciation lattice. Their experiments showed that with an increase of regression classes added to the hierarchy, the PER was more and more reduced and got close to the performance obtained with surface form trained acoustic models.

5.2.3 Benefits and limitations of SLPM

A benefit of SLPM is the possibility to model pronunciation variations with a higher granularity through the creation of hybrid acoustic models than conventional phone-based methods that are limited by the size of their phone inventory. Another benefit is that the lexicon need not be expanded with new pronunciation variants, which not only eliminates extra computation time involved by the increased size of the lexicon and of the recognition network, but also does not increase lexical confusion.

However, SLPM has also its limitations and drawbacks. First of all, even though the number of parameters does not increase much since Gaussians are shared, hybrid phone models have a higher number of Gaussians than the phoneme and phone models they originate from, which still involves some extra computation time during decoding. Next, although lexical confusion is reduced, *acoustic* confusion may be increased because several acoustic models share the same Gaussians (even though their mixture weights are generally different). Since SLPM provides higher modeling granularity and resulting hybrid models are still acoustically different from original phoneme and phone models, acoustic confusion will hopefully not be as high as lexical confusion.

5.2.4 Characteristics of a dynamic SLPM

This chapter will investigate whether a dynamic approach could help to decrease acoustic confusion and the WER. Dynamic SLPM is different from the conventional (static) version due to the following points:

1. Sharings of Gaussian densities are processed *during recognition* and they vary from one utterance to another, while they are still governed by a pronunciation model.

2. Even if a phoneme /b/ may be realized as a phone [s], it does not necessarily create a global hybrid [b_s] model applicable to all lexical entries containing /b/, but only to

some of the entries.

3. Even if the phoneme /b/ of two lexical entries may share the Gaussian densities of [s], a specific hybrid [b_s] model may be created for each entry.

In order to compare the static and dynamic approaches, both techniques will be described in the next sections.

5.3 Static SLPM

Similar steps to those reported by Saraçlar et al. [124] were followed to first build a static SLPM-based system. It was assumed that Gaussian mixtures were used as emission densities, and each density in the original system was supposed to belong only to a single state. The following steps were applied during the training phase:

1. The phonemic transcription of a sentence built from the lexicon was aligned with the manually labeled phonetic transcription of the same sentence. We used the same phoneme-to-phone alignment based on the phonetic feature distances described in section 4.6.2. State-to-state correspondences were directly deduced from the alignment results.

2. From the alignments above, the probability of a state b in the canonical transcription to be aligned to a state s in the surface form transcription was estimated: $P(s \mid b) \approx \frac{Count(s,b)}{Count(b)}$

3. Any pair (s,b) with $Count(s,b)$ less than a threshold T_{count} or $P(s \mid b)$ less than a threshold T_{prob} was pruned. Probabilities of the remaining pairs were renormalized.

4. The new output distribution of state b was estimated. This was a sum of all the mixtures of states s remained after pruning. The probabilities $P(s \mid b)$ were used to compute the new mixture weights:

$$P'(o|b) = \sum_{s:P(s|b)>0} P(s|b) \sum_{i=1}^{N_s} w_{i,s} \mathcal{N}(o; \mu_i, \Sigma_i) \tag{5.1}$$

   $P'(o \mid b)$ is the new output distribution of state b, and N_s, $w_{i,s}$ and $\mathcal{N}(o; \mu_i, \Sigma_i)$ are the number of mixtures, the i-th mixture weight and the i-th distribution of state s in the original system, respectively. Note that as state-to-state alignments were inferred from phoneme-to-phone alignments, $P(s \mid b)$ was the same for all states of the same model.

5. Parameters of the new acoustic models and weights were re-estimated with the Baum-Welch algorithm.

An example of output distribution (before the model re-estimation step) is shown in Figure 5.2 when starting with single Gaussian for both the phoneme /b/ and phone [s]. ω_b and ω_s are the original mixture weights of /b/ and [s] respectively (both equal to 1 since there is only one Gaussian in each original distribution), and $P(b \mid b) \cdot \omega_b$ and $P(s \mid b) \cdot \omega_s$ are the weights of the output distribution after application of SLPM.

The steps above were applied to all training sentences and the modified phonetic models were used for recognition.

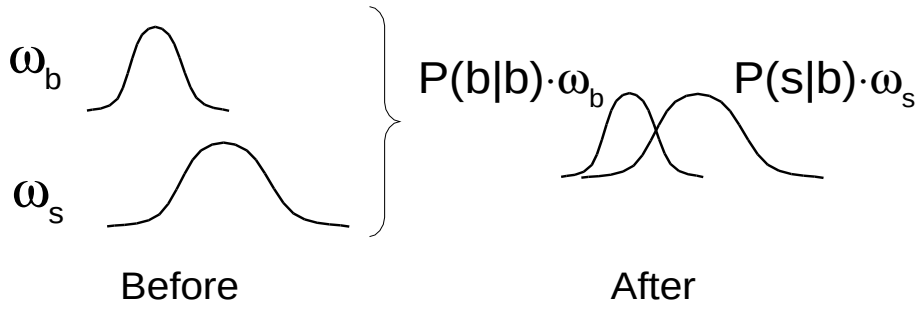

Figure 5.2: Example of output distribution using SLPM

5.4 Dynamic SLPM

5.4.1 Overview

In dynamic SLPM, decision of sharing Gaussian densities between two acoustic models relied on the detection of phonetic features. The following steps depicted in Figure 5.3 were respected for each utterance of the *testing* set:

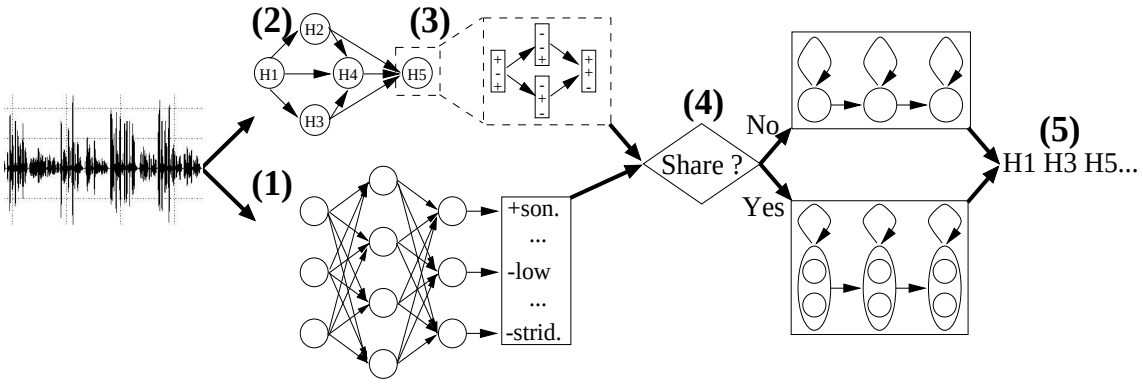

Figure 5.3: Overview of dynamic SLPM

1. Some phonetic features were first extracted from the input speech on a frame-by-frame basis using an artificial neural network.

2. Independently, a baseline HMM system was used to apply a first recognition pass to the same input speech and to generate a lattice of the most likely word hypotheses with their time boundaries.

3. For each hypothesis, a procedure (explained in section 5.4.4) mapped the word to a graph of phonetic features.

4. The graph in step 3 was compared to the phonetic features returned by the neural network in step 1 over the given word's time interval. Depending on how much the features differed from each other (explained in section 5.4.5), some Gaussian mixtures were eventually shared between HMMs.

5. The hybrid models were added to the original set of HMMs for a second pass recognition.

Each step will be explained thoroughly in the next subsections.

5.4.2 Extraction of phonetic features from speech

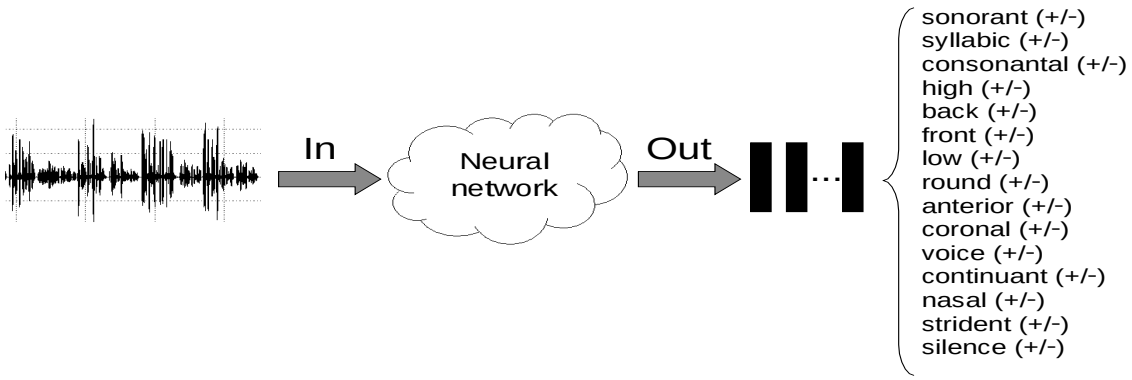

Figure 5.4: Detection of phonetic features from speech using an artificial neural network

As in the previous chapter, an artificial neural network (ANN) mapped a set of acoustic parameters to phonetic features, frame-by-frame. Again, the SPE system [25] was selected as the feature set. However, a different version of phone-features conversion table (proposed by Brondsted [17]) was employed so that each phone was mapped to a unique combination of phonetic features. The list of features used is shown in Figure 5.4. The complete list of phones and their corresponding features can be found in appendix B.2.

5.4.3 First recognition pass

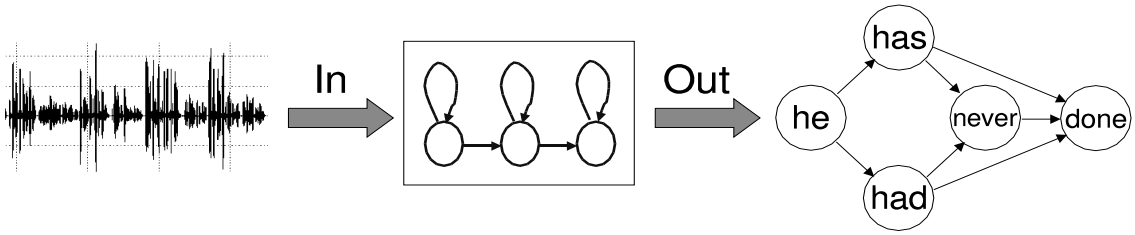

Figure 5.5: Generation of a lattice of word hypotheses from speech

An HMM-based ASR system applied a first recognition pass to the input speech and generated a lattice of the most likely word hypotheses, as illustrated in Figure 5.5. Only phoneme models associated with these words were eventually subject to Gaussian sharings. Start and end times of each word were retained for later steps. To reduce processing time, if a word was located at several places in the lattice, only the time interval of the hypothesis with the highest acoustic likelihood was retained.

5.4.4 From word hypotheses to phonetic features

Each word hypothesis was mapped to a graph of phonetic features, which helped later to decide whether models representing this word could be transformed or not, and if so which ones. The graph was constructed thanks to the following steps (an example is given for the word “had” in Figure 5.6) :

1. The word’s canonical transcription was extracted from the lexicon.

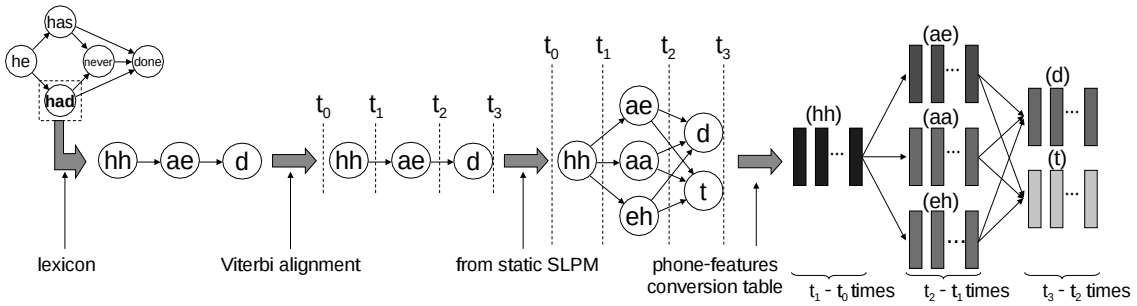

Figure 5.6: Creation of a graph of phonetic features from a word hypothesis

2. A Viterbi alignment between the transcription and the speech waveform over the given word's time interval was applied in order to get segmentation points of its phoneme constituents.

3. Each phoneme was mapped to a set of phones obtained by the procedure described for the static SLPM in section 5.3. A graph of possible phones was therefore generated. To simplify the process, phones were assumed to share the same segmentation points as their corresponding phoneme.

4. Each phone in the graph was mapped to its corresponding vector of phonetic features using a phone-features conversion table. The vector was duplicated as many times as there were frames attributed to this phone.

5.4.5 Comparisons of phonetic features

The graph of feature vectors generated in the previous step was compared to the sequence of feature vectors returned by the neural network (cf. section 5.4.2). Comparisons were done separately for each sequence of feature vectors in the graph over the time intervals given by their segmentation points. Each comparison consisted in evaluating the same *measure of similarity* (MoS) described in the previous chapter (section 4.6.3) between two phones A and B, represented here by two sequences of phonetic feature vectors:

$$\log MoS(A,B) = \frac{1}{N}\sum_{n=1}^{N} \log MoS(\overrightarrow{a_n}, \overrightarrow{b_n}) = \frac{1}{N}\frac{1}{M}\sum_{n=1}^{N}\sum_{m=1}^{M} [1 - |targ(a_n^m) - act(b_n^m)|] \quad (5.2)$$

where:

- A and B are the sequences of phonetic feature vectors (A is from the graph, B is from the output of the ANN) compared over an interval of N frames
- $\overrightarrow{a_n}$ and $\overrightarrow{b_n}$ are single feature vectors belonging to A and B respectively
- $targ(a_n^m)$ is the target value for the m-th feature of the vector $\overrightarrow{a_n}$
- $act(b_n^m)$ is the activation value for the m-th feature of the vector $\overrightarrow{b_n}$, returned by the ANN for the n-th frame.
- M is the number of features per feature vector

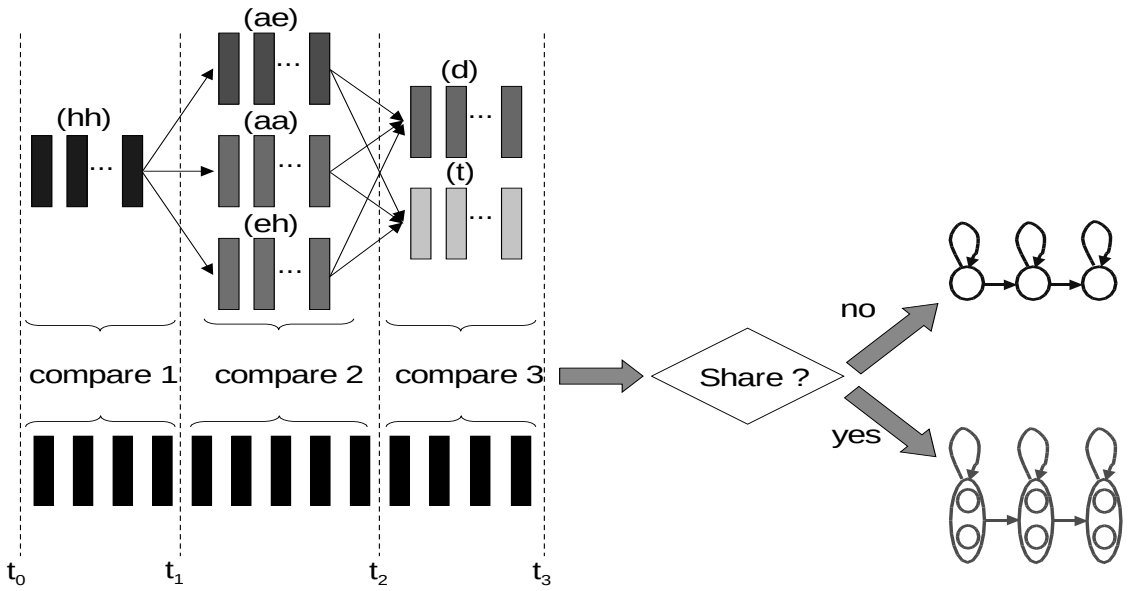

Figure 5.7: Comparisons between a graph of phonetic feature vectors and a sequence of phonetic feature vectors extracted from speech

For each phoneme represented by a set of sequences of feature vectors in the graph (*e.g.*, in Figure 5.7, sequences representing [ae], [aa] and [eh] are associated with the phoneme /ae/ of the word "had"), the sequence (or equivalently the represented phone) with the maximum MoS was retained. Once this was done with all phonemes, a path going through each selection represented the best path $\mathcal{P}$. The MoS of the path is given by :

$$\log MoS(\mathcal{P}) = \frac{1}{N} \sum_{p \in \mathcal{P}} N_p \log S(A_p \rightarrow B_p) \tag{5.3}$$

where N_p ($\sum_{p \in \mathcal{P}} N_p = N$) is the number of frames covering the selected phone p. Gaussian sharings were allowed for a word only if $MoS(\mathcal{P})$ was above a threshold T_{match} (fixed to 0.5 in our experiments). Moreover, a phoneme's model could share Gaussians of an alternative distinct phone only if the MoS of this phone was higher than both the threshold T_{match} and the self-MoS of the phoneme (*i.e.*, the MoS of a phoneme being mapped to itself). By default, a phoneme was at least always mapped to itself.

As an example, suppose that a phoneme /aa/ may be realized as the following phones with their respective MoS in parentheses: [aa](0.26), [ah](0.13), [ao](0.40) and [ax](0.64). The phone [ah] is not a candidate for sharing because its MoS (0.13) is both lower than the self MoS ([aa], 0.26) and T_{match} (0.50). [ao] is not a good candidate either because its value is still lower than T_{match}. Only [ax] satisfies both conditions ($0.64 > 0.26$ and $0.64 > 0.50$) and shares its Gaussian mixtures with [aa]. Probabilities of associations $P(s \mid b)$ used to compute the new mixture weights in equation (5.1) were estimated from the selected MoS. In this example, we have:

$$P(aa \mid aa) = 0.26/(0.26 + 0.64) \cong 0.29$$
$$P(ax \mid aa) = 0.64/(0.26 + 0.64) \cong 0.71$$

5.4.6 Second recognition pass

Once all hypotheses were processed and the appropriate transformations applied, the new HMM models were added to the original set of models for a second recognition pass. The

lexicon was also updated to take account of the changes. Note that two symbols found in the new lexicon and associated with the same phoneme could refer to different models (for example, "bar → [b aa1 r]" and "car → [k aa2 r]" referred to two different output distributions of the phoneme /aa/) since their respective MoS were generally different.

5.5 Experiments and basic results

5.5.1 Database and recognition tools

The material used for the SLPM experiments was the same as presented in the previous chapter: TIMIT [51] as the database, HTK [158] as the HMM-based ASR system and the NICO toolkit [107] as the ANN to detect phonetic features from speech. More detail can be found in section 4.7.

5.5.2 Baseline system

A baseline HMM system was built using the reduced set of 40 phones proposed by Brondsted [17] (shown in appendix B.2) in order to associate each phone to a unique set of SPE features used in the experiments. A "silence" and "short pause" models were also added. All models had the standard three left-to-right states with no skips, except for "short pause" that had only one state tied to the center state of "silence" and for which skip of the model was allowed. The system was trained using 39 MFCC coefficients (12 static + 1 normalized log energy, 13 Δ, 13 $\Delta\Delta$) and the manually labeled transcriptions of TIMIT. More information about the training procedure can be found in appendix C.1.

The monophone models after training had 10 Gaussian mixtures per state. A back-off bigram (same as in the previous chapter) was built from all distinct sentences of TIMIT for evaluation. Evaluated with the core test set of TIMIT, the system achieved a 14.8% Word Error Rate (WER) (85.2% accuracy).

5.5.3 Phonetic feature recognition results

For compatibility with the trained HMM system, the ANN was also trained using the same set of 40 phones found in [17] with their corresponding vectors of SPE features (cf. appendix B.2). The original phone-features conversion table was a ternary version that included non-relevant features, but training with ternary targets led to significantly lower results (around 40% "all correct" rate), so we preferred to use a binary version for training purposes only (the binary version can also be found in appendix B.2). The topology of the ANN was identical to the description of section 4.7.3 in the previous chapter, except that one more unit (and related links) was added to the output layer to represent the higher number of SPE features compared to the previous version (14 SPE + 1 silence, instead of 13 SPE + 1 silence). The training procedure was identical to the one reported in section 4.7.3. Since the original 62 phones were reduced to a set of 40 phones, reference phone labels provided by TIMIT were modified accordingly. When a diphthong was mapped to two monophthongs (*e.g.*, [iy] mapped to [ih] + [y]), the time interval of the diphthong given by the original labels was equally divided by two to associate them with each monophthong.

Comparisons between recognized features and those derived from the reference phone tran-

scriptions of TIMIT led to the results in Table 5.1, given in percentage of frames correct on the cross-validation set. As expected, performance was similar to the one reported in the previous chapter (cf. Table 4.5).

Feature	Frames correct (%)	Feature	Frames correct (%)
sonorant	95.4	round	92.3
syllabic	89.5	anterior	90.0
consonantal	91.0	coronal	87.7
high	88.0	voice	89.4
back	92.3	continuant	91.4
front	93.0	nasal	97.7
low	91.8	strident	96.9
		silence	98.3
Average	**92.3**	**All correct**	**52.8**

Table 5.1: Frame-level classification results with SPE features

5.5.4 Results with static SLPM

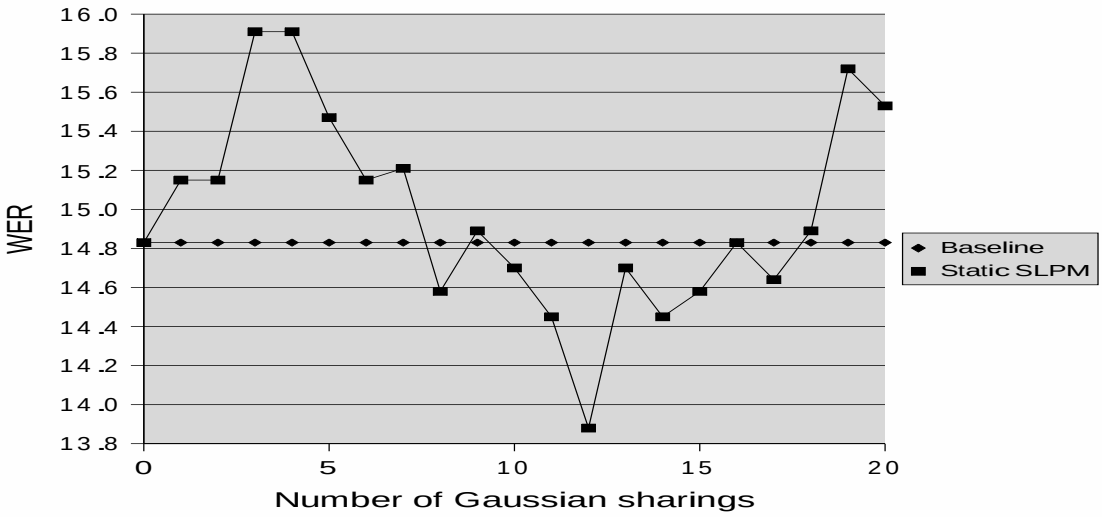

Figure 5.8: Evolution of the WER with respect to the number of Gaussian sharings when using static SLPM

The sequence of steps described in section 5.3 was applied to all SX and SI training sentences of TIMIT. At the end of the procedure, we obtained all possible pairs of states (s,b) with their probabilities of alignments $P(s \mid b)$. Instead of fixing the thresholds T_{count} and T_{prob} (used to prune unreliable pairs), we preferred to set T_{count} to zero (no "count" threshold), and let the probability threshold T_{prob} be variable, in order to see the evolution of WER with respect to the number of sharings. Namely, sharings were progressively added one after another, starting with the pair with the highest probability of association ($P(s \mid b)$). As a consequence, the average number of Gaussians per state increased from 10 to 14.8. The graphic in Figure 5.8 shows the results obtained with the 20 first sharings. We notice that the new WERs vary around the baseline WER. Slight improvements were observed around 12 Gaussian sharings with the best result at 13.9% WER (86.1% accuracy), but they were not statistically significant. It seems that any improvement brought by sharing Gaussian densities

may have been counterbalanced by an increase of acoustic confusion between phonetic models. We did not get any more improvement by further increasing the number of sharings.

5.5.5 Results with dynamic SLPM

HTK used the token passing algorithm [159] to perform an N-best recognition and to generate a lattice of word hypotheses per utterance. Maximum three tokens were admitted in each network node at each iteration of the algorithm. Thresholds T_{count}, T_{prob} and T_{match} were fixed to 10, 0.01 and 0.5 respectively. Short words (such as "a", "or", ...) were excluded from sharings because the corresponding time intervals returned by the HMM system at the first pass were often wrong[2]. Results are given in Table 5.2. Performance of the dynamic approach is close to the best result obtained with the static approach, but the corresponding improvement is still not statistically significant: only a 5.4% relative reduction in WER compared to the baseline system could be obtained.

Method	WER
Baseline	14.8
Static SLPM	13.9 to 15.9
Dynamic SLPM	14.0

Table 5.2: Recognition results with static and dynamic SLPM

Several points may explain the lack of significant improvement with both the static and dynamic SLPM. First, TIMIT, which is a carefully read speech database, does not contain as much pronunciation variation as spontaneous speech databases; this may partly explain the slightly better improvement obtained by Saraçlar et al. [124] on Switchboard.

Second, several components of the dynamic SLPM were prone to errors that could have been propagated to other parts of the system. Some of the components are:

Detection of phonetic features : we already mentioned in the previous chapter the need to detect phonetic features more reliably. This is a general problem shared by many researchers working in this field of interest.

First pass recognition : if a lattice of words output by the first recognition pass does not contain the correct word(s), it (they) won't be processed by dynamic SLPM and there is little chance it (they) will be recognized successfully during the second pass. An increase of the lattice size would of course insure a higher inclusion rate of correct words, but also of wrong words and would increase processing time.

Time intervals of word hypotheses : when a word hypothesis was located at several places of the lattice, the interval with the maximum time normalized likelihood was selected. A more elaborate approach would be needed to more reliably estimate intervals of correct (and especially short) words.

Graph of phonetic features : sequences of feature vectors referring to the same phoneme shared the same time segmentation points. A more accurate approach with different segmentation points could be helpful.

[2]This problem is similar to the difficulty of searching short word transcriptions using the pronunciation search algorithm in the previous chapter (section 4.8.4).

Finally, SLPM can only account for phoneme substitutions: when a phoneme is frequently realized as a given phone, it shares the Gaussian densities of the latter to partially acquire its acoustic properties. However, in the case of a deletion or an insertion, it is not possible to establish any relationship between a valid phoneme and a valid phone, hence SLPM cannot be applied.

This is the reason why the next section is dedicated to the incorporation of deletions and insertions, based on decision trees. It will compare the performances between trees and phonetic features when they are separately incorporated into the dynamic SLPM framework. Furthermore, it will check whether both techniques (trees + features) are complimentary and can further increase performance.

5.6 Modeling of deletions and insertions

In order to model deletions and insertions (and more generally pronunciation variations), we decided to rely on a technique already applied in pronunciation modeling. Among the techniques reported in the literature, decision trees kept our attention and our preference because:

1. Hierarchical organization of decision trees makes them suitable to generalize prediction of pronunciation variations to unseen contexts.
2. The set of features composing the trees is easily modifiable and expandable.
3. A set of powerful tree-based classification algorithms exist and were successfully applied in pronunciation modeling (*e.g.*, Riley and Ljolje [119]).

Among the possible choices of tree-based classification algorithms, we decided to use the CART (classification and regression trees) [16] methodology, which is known to be effective and well suited with limited amounts of data. The next subsection gives a general description of the CART algorithm.

5.6.1 The CART algorithm

CART has become a commonly used method to build decision trees. We suppose that a set of samples is available. Each sample is associated with a vector of features that characterizes it, and it is supposed to belong to a certain class. The objective is to build a tree that "best" (this term will be specified later) distributes the samples into their classes - represented by the leaves of the tree. For this purpose, a list of rules that split the set of samples must be available. In CART, splitting rules are questions about the sample features and always wait a "yes" or "no" as an answer (*e.g.*, "Is age $\leq$ 18 ?"). So the task of the CART algorithm is, starting from all samples in the root node, to recursively split them in two subsets using one of these rules and to distribute them to the nodes of the next hierarchy level.

Given this task, a first point to specify is to decide in which order the rules should be applied. At each hierarchy level, CART conducts an exhaustive search by trying all available rules in order to see how each of them splits the current set of samples. The rules are ranked according to a certain quality-of-split criterion and the best rule from the list is used to split the data in two sub-partitions. Several criteria exist, but the Gini rule is commonly used in

CART. It quantifies the misclassifications resulting from the associations of a given sample of a node to a class s_i while it actually belongs to another class s_j:

$$Gini = \sum_{\forall i,\forall j,i\neq j} p(s_i)p(s_j) = \sum_{\forall i,\forall j} p(s_i)p(s_j) - \sum_{\forall i} p(s_i)p(s_i) = 1 - \sum_{\forall i} p(s_i)^2 \qquad (5.4)$$

A purity measure Φ is simply obtained by $\Phi = 1 - Gini$: the higher this value, the purer a partition is. It was shown in [16] that this purity measure never decreases and most often increases at each split. Other quality-of-split criteria exist, for example based on entropy (see section 5.6.2).

A second point to specify is when to stop the algorithm. Early tree classifiers continued to split samples until a stopping criterion was met, for instance when the number of samples in a node fell below a certain threshold or when the purity measure did not increase more than a certain value with a given split. CART does not stop in a middle point, because still important information can be found by exploring further splits; so it continues until the tree cannot be grown any further, that is, until the remaining rules cannot split any set of samples in two. Once this maximal tree has been grown, a set of smaller trees is created from it by pruning away some branches. These subtrees must be the smallest trees that minimize a certain *misclassification rate* or *complexity cost* R_α, where α is a certain *complexity coefficient*. It results a parameter family of subtrees, one for each given range of α. In order to define the best subtree, CART determines which is the best range of α using a separate held-out data.

Alternatively when little data is available, CART applies an n-fold cross-validation scheme. The training data is divided in N parts roughly equal in size. $N-1$ parts are used to generate a maximal tree and corresponding set of subtrees, and the remaining part is used as a cross-validation set to determine an estimate of misclassification rate R_α. Then, another $1/N$ is used for cross-validation and the remaining parts to grow another tree and set of subtrees. The process is repeated until each part has been used once as a cross-validation set. Finally, all estimates of misclassification error rates are combined to give an average value for each range of the complexity coefficient α, which determines the best subtree derived from the entire training set.

5.6.2 Building of decision trees

A tool called *Wagon* [143] developed by the Edinburgh university was used to build decision trees using the CART algorithm. A separate tree was built per phoneme to predict a list of realizable phones given phonetic features of its left and right phonemic contexts. The list of features was based on the example provided with HTK for tree-based clustering, but adapted for the phone inventory used in experiments; see appendix D for an exhaustive list of features and associated phones. Data to build the trees was created from phoneme-to-phone alignments between canonical and manually labeled transcriptions of all SX and SI training sentences of TIMIT. A special symbol '?' was included in these alignments to account for deletions and insertions. 90% of the training set were used to build the maximal tree according to the CART algorithm and the remaining 10% were reserved as held-out data to prune back the maximal tree and to determine the best subtree. At each step of the building process, the Wagon program chose the question that led to the least impure sample sets. The impurity of a sample set was defined as the entropy of the sample set times the number of samples:

$$-\sum_{\forall i}[p(s_i)\log(p(s_i))]\cdot N \tag{5.5}$$

where $p(s_i)$ is the probability distribution of a class s_i and N is the number of samples in a node. The number of samples was included in the expression to encourage the creation of larger partitions.

5.6.3 Usage of decision trees

The created decision trees were incorporated into the dynamic SLPM framework. However, for comparison purposes, a first preliminary version was designed that did not rely on the detection of phonetic features, but solely on decision trees, as described by the following steps:

1. A first recognition pass generated a lattice of word hypotheses from speech.
2. For each distinct word, the hypothesis with the highest time normalized acoustic likelihood was selected. Decision trees were applied to expand the corresponding canonical transcription to a graph of phones with associated probabilities (see comments below).
3. Based on comparisons of phone probabilities, some Gaussian mixtures were eventually shared between HMMs (see comments below).
4. The hybrid HMMs were added to the original set of models for a second pass recognition.

Comments on step 2:

In the original configuration, a counting and pruning procedure was implemented in static SLPM to map a phoneme /b/ to a set of phones [s] (see section 5.3); this procedure was also applied in dynamic SLPM to create a graph of phones from the canonical transcription of each distinct word hypothesis (see section 5.4.4). In the new configuration, this method was replaced by the decision trees: for each phoneme, the associated tree was explored by asking questions about phonetic features of the left and right contexts of the phoneme, then the corresponding leaf of the tree contained the set of realizable phones with their associated probabilities. A phoneme was considered as deleted if the [?] (= deletion) phone was ranked first in this set. No phone insertion was considered at this stage. Probabilities associated with phones were used to determine new mixture weights in the dynamic SLPM framework (see comments for step 3).

Cross-word pronunciation modeling was also taken into account: for each boundary phoneme of a word hypothesis, all last phonemes of predecessors or first phonemes of successors of the hypothesis in the lattice were considered as possible contexts to generate a global set of realizable phones. If a same phone was predicted for several contexts, the average value was calculated from the probabilities found in the corresponding leaves.

Comments on step 3:

In the original configuration, measures of similarity (MoS) were calculated between sequences of phonetic features deduced from phones (using a phone-features conversion table) and sequences of same features directly detected by an ANN from speech. Depending on

their similarities, some phonetic models eventually shared their Gaussian densities (see section 5.4.5). In the new configuration, the MoS were simply replaced by the probabilities returned by the decision trees for each phone. Namely, if the probability of a phoneme /b/ to be realized as a distinct phone [s] was higher than the probability of realizing it as itself ([b]), the acoustic model of /b/ was allowed to share the Gaussian densities of [s].

5.6.4 Results with decision trees

A first evaluation was performed to insure that usage of decision trees led to higher phone accuracy than canonical transcriptions. Their performance was directly evaluated in the updated dynamic SLPM method described in the previous subsection by respecting the following steps for each reference word of each test sentence.

1. The reference word was searched in the lattice of hypotheses generated by the first recognition pass.
2. If the word was found in the lattice, decision trees had been applied, according to the updated dynamic SLPM framework, to the phonemic transcription and contexts of its best hypothesis to generate a graph of phones and associated probabilities. Only the best sequence of phones given by the maximum probability path of the graph was considered. If the word was not found, its canonical transcription was used instead.
3. The best sequence of phones was aligned to the corresponding reference phonetic transcription and the number of alignment errors was counted.

The final result was compared to the baseline accuracy obtained by aligning canonical transcriptions of correct words to the corresponding reference transcriptions. The result was also compared to the accuracy obtained with the original dynamic SLPM configuration that relied on phonetic features only. Evaluation of the original system followed the same steps as above, except that:

- Any graph of phones given a phonemic transcription was obtained by the counting and pruning procedure explained in section 5.3 and not by decision trees.
- Phoneme deletions were not modeled (the counting and pruning procedure was originally designed for the static SLPM, which only handled substitutions).
- Probabilities associated with phones to find the best path in graphs were not given by decision trees, but by measures of similarity (MoS) from comparisons of phonetic features.

Decision trees were also evaluated for word recognition by following the procedure described in the previous subsection and were compared to the baseline and original dynamic SLPM performances. The resulting phone and word error rates are shown in Table 5.3[3]. The system based on phonetic features alone did not perform better than canonical transcriptions. The number of insertions and deletions remained almost the same[4] since they were not modeled.

[3]The baseline PER is different from the result reported in the previous chapter (section 4.8.2), because phone inventories and phone-features conversion tables used are also different.

[4]The slight differences of results between the two systems are apparently due to possible alternative alignments leading to the same number of errors but of different types for some sentences.

The number of substitution errors was on the other hand much higher. This result was however a bit expected since in the previous chapter (section 4.8.2), we had already noticed that pronunciation networks built from phonetic features had not performed better than canonical transcriptions in terms of phone accuracy. Nevertheless, we had also noticed in the previous chapter that combination of feature-based pronunciation networks and canonical transcriptions had led to significantly higher performance. The next section will also study a combination scheme, this time with decision trees and phonetic features.

Method	PER	Subst.	Ins.	Del.	WER
Baseline	21.9	911	364	225	14.8
Phonetic features	23.2	986	373	233	14.0
Decision trees	20.6	836	284	294	15.1

Table 5.3: Phone and word recognition results with phonetic features or decision trees incorporated into the dynamic SLPM framework

Application of decision trees did reduce the PER compared to the baseline system, although not by much (only 5.9% relative reduction). A more detailed analysis of the errors showed that the number of substitutions was reduced, but decision trees predicted substantially much more deletions than it was necessary, so that the reduction of phone insertions was almost counterbalanced by an increase of phone deletions. This nevertheless slight improvement in phone accuracy did not however influence the WER and even slightly degraded the performance, suggesting perhaps that Gaussian sharings governed by decision trees were too often applied and increased acoustic confusion. The next subsection will study how these issues could be addressed.

5.6.5 Combination of decision trees and phonetic features

Decision trees and phonetic features could contain partially complementary information, hence it was desirable to check whether their combination could yield additional improvement. The general method with both components can be described by the following steps (replacements and additions compared to the dynamic SLPM of section 5.6.3 are in bold):

1. **Some phonetic features were extracted from the input speech on a frame-by-frame basis using an artificial neural network (ANN).**

2. Independently, a first HMM recognition pass generated a lattice of word hypotheses from the same speech.

3. For each distinct word, the hypothesis with the highest time normalized acoustic likelihood was selected. Decision trees were applied to expand the corresponding canonical transcription to a graph of phones with associated probabilities. **Phone deletions were further restricted by comparisons of phonetic features deduced from the graph with those returned by the ANN in step 1** (see comments below).

4. **Based on comparisons of tree-based phone probabilities and of phonetic features combined, and based on the presence of reliable feature-based hypotheses** (see comments below), some Gaussian mixtures were eventually shared between HMMs.

5. The hybrid HMMs were added to the original set of models for a second pass recognition.

Comments on step 3:

In the previous configuration, a phoneme deletion occurred whenever it was considered the most probable by the decision trees. We noticed that this decision resulted in too many deletions. Therefore, the following approach was adopted to further restrict the number of deletions using phonetic features:

- Even if a phoneme deletion was given as the most probable, it was ignored and the remaining possibilities found in the corresponding tree leaf mapped the phoneme to a set of realizable phones.
- For each phone candidate, its phonetic features were obtained using a phone-features conversion table and were compared to the features returned by the ANN over the same time interval[5] to give a measure of similarity (MoS) for each phone.
- If at least one phone had a MoS above a threshold T_{match} (= 0.50 in our experiments), the phoneme was considered as *not* deleted. A deletion was accepted only when both a deletion was ranked first by decision trees *and* no MoS for the alternative realizable phones was above T_{match}.

Comments on step 4:

In the original configuration, decision of sharing Gaussian densities between acoustic models was governed by the MoS obtained from comparisons of phonetic features. In the updated configuration with decision trees, probabilities of predicted phones returned by the trees were used instead for the same purpose. In this configuration, a combination scheme was adopted by taking the average of MoS and tree-based probabilities. Namely, if a phoneme /b/ could be realized as a phone [s], the final associated probability was:

$$P_{final}(s \mid b) = \frac{MoS(b,s) + P_{tree}(s \mid b)}{2} \tag{5.6}$$

This probability was also used to determine the initial new mixture weight of any Gaussian-shared output distribution, following the method given by the example of section 5.4.5. Moreover, the concept of feature-based *reliable hypotheses* seen in the previous chapter was reintroduced to further restrict the number of Gaussian sharings. In contrast with the word hypotheses mentioned in this chapter, feature-based hypotheses were built by binding several successive frames whose associated phonetic features referred to a same phone (please refer to section 4.5 for more details). Conditions for a hypothesis to be considered as reliable were originally defined in section 4.5.4. The conditions applied here were slightly relaxed compared to the previous version in order to encourage the creation of more reliable hypotheses (and consequently to further restrict the number of Gaussian sharings, see below). A hypothesis was considered as reliable if at least R_{min} (= 2 in our experiments) of its frames respected the following conditions:

- The phone associated with the hypothesis was ranked first, or at least second behind a transitional phone.
- Difference of penalty score between this phone and the next ranked phone was equal or above a certain threshold (= 0.75 in our experiments).

[5] Segmentation points of phones were obtained by Viterbi alignment of the corresponding word to the speech segment associated with its best hypothesis, see section 5.4.4.

Ranking procedure and determination of penalty scores were explained in section 4.5.2, and the notion of transitional phone was introduced in section 4.5.3.

Reliable feature-based hypotheses further restricted the number of Gaussian sharings in the following way:

- For a given word hypothesis it was checked whether its time interval contained one or more feature-based reliable hypotheses. At least 50% of its frames had to be in the interval to accept the inclusion of a reliable hypothesis.
- The phoneme constituent of the word hypothesis whose time interval most overlapped the interval of the reliable hypothesis was selected.
- If the phone represented by the reliable hypothesis was among the realizable phones returned by the decision trees given the selected phoneme and its contexts, the reliable hypothesis was considered as present, otherwise as absent.
- If all feature-based reliable hypotheses could find a match among the realizable phones, the word hypothesis matched well the detected phonetic features and was retained for possible Gaussian sharings. Otherwise, if at least one reliable hypothesis remained, the word was considered as a bad match and its phonemes were not allowed to share any Gaussian density with other models.

Alignment of reliable hypotheses relative to time intervals of word hypotheses and their phonemes was rather rudimentary and certainly leaves some room for improvement. A further restriction was applied to short word hypotheses, whose time intervals could still not be reliably estimated: phonemes of a short word were allowed to eventually share Gaussian densities only if the word was in the best sequence returned by the first Viterbi recognition pass. Any short word satisfying this condition was then treated like any other word.

5.6.6 Results with combination of decision trees and phonetic features

Hypothesis identification accuracy

A first experiment measured how accurately dynamic SLPM was able to distinguish the correct from the wrong hypotheses for Gaussian sharings. In this combined version of dynamic SLPM, a word hypothesis was considered as a valid candidate for Gaussian sharings if:

1. At least half of its phonemes had at least one associated phone MoS above the fixed threshold T_{match} (= 0.5 in our experiments).
2. All feature-based reliable hypotheses contained in the word's time interval found a match (see section 5.6.5, step 4).

Two types of errors were used to evaluate the accuracy: a *false alarm* rate that measured how frequently a wrong word hypothesis was considered as a valid candidate, and a *miss* rate that measured how frequently a correct word hypothesis was considered as a non-valid candidate. These errors were compared to the errors obtained with the previous configurations, namely with decision trees alone and with phonetic features alone. Results are given in Table 5.4.

Method	False alarms (%)	Misses (%)
Decision trees	85.6	3.6
Phonetic features	9.9	50.3
Trees + features	27.1	17.5

Table 5.4: False alarm and miss rates with dynamic SLPM using decision trees only, phonetic features only and combination of both trees and features

Results with decision trees can be considered as the baseline performance: since no phonetic features to compare with were detected from the signal, all word hypotheses were considered as potential candidates for Gaussian sharings, which generated of course a lot of false alarms. In other words and according to the results, 85.6% of selected hypotheses in lattices represented not uttered words. The 3.6% miss rate means that this percentage of correct words were not in the lattices generated by the first HMM recognition pass and could not be selected as potential candidates. These two values were therefore respectively the highest false alarm rate and the lowest miss rate a system could achieve given the lattices. Results with the original configuration with phonetic features show a reversed behavior: while the number of false alarms is radically smaller, the percentage of misses also substantially increased with more than one correct word out of two rejected on average. On the other hand, the combined system with both decision trees and phonetic features led to a more acceptable compromise between false alarm and miss rates.

Phone and word recognition accuracy

A second experiment evaluated how the combined system performed in terms of phone and word accuracy. Experiment settings were basically similar to the description given in section 5.6.4, however adapted to the new configuration to respect the modifications described in the previous subsection. Results are given in Table 5.5. Performance of the combined system was perceptibly better in terms of phone accuracy: the number of substitutions further decreased and although the number of deletions still increased a bit, it was better compensated by the diminution of insertions (the decrease of insertions was approximately 2.8 times higher than the increase of deletions for the combined system, vs. 1.2 times for the tree-based system). The final relative reduction in PER compared to the baseline system (12.3%) was small but statistically significant. But surprisingly, even this higher performance in phone accuracy did not have any impact on the WER.

Method	PER	Subst.	Ins.	Del.	WER
Baseline	21.9	911	364	225	14.8
Phonetic features	23.2	986	373	233	14.0
Dec. trees	20.6	836	284	294	15.1
Trees + features	19.2	756	325	239	15.1

Table 5.5: Phone and word recognition results with phonetic features only, decision trees only and combination of both trees and features incorporated into the dynamic SLPM framework

It is difficult to understand why a significantly lower PER did not yield better word recognition performance. Saraçlar et al. [124] also experienced such contradictory behavior with their phone-level pronunciation modeling techniques. They believed that it was possibly due to lexical confusion, thus rendering identification of words from phones more difficult. Similarly, it is possible that acoustic confusion involved to a certain extent by dynamic SLPM

had also an influence in a similar way. It is also hard to understand why the system with the worse PER led to the best WER. It is in this sense interesting to note that Kessens and Strik [78] observed similar trend: in their own experiments, context-dependent HMMs, which had the lowest WER, did not generate better phonetic transcriptions than context-independent HMMs with the highest WER. Saraçlar et al. [124] argued that a too high number of parameters may constitute a limitation and slightly improved their phonetic transcriptions by using simpler acoustic models. We cannot however make a direct relationship between their observations and our own experiments because all acoustic models used here were monophones.

Modeling of insertions

The system that combined both decision trees and phonetic features predicted phoneme substitutions and deletions, but not insertions. An additional test was therefore included to model insertions as well. In the last configuration, if a reliable hypothesis (reminder: this is a feature-based hypothesis that represents a phone) was included in a word's time interval, but its phone label matched neither the best time aligned phoneme of the word nor its realizable phones, the word was considered as a bad match and no SLPM was allowed with its phonemes. In the new configuration, a missing reliable hypothesis was interpreted as a possible insertion of the phone either on the left or on the right of the considered phoneme of the word. If, under this situation, the decision trees also predicted the missing phone as the best insertion choice for at least one side (the side with the highest probability was chosen if both sides were possible), then the insertion was accepted and the original phonemic transcription modified accordingly. Results are given in Table 5.6. The new approach was not successful and provoked a substantial increase of insertion errors. More accurate methods need therefore to be experimented, for instance by modeling insertions separately for each phoneme (*e.g.*, as done by Humphries [67] and in the next chapters) instead of building a global insertion tree.

Method	PER	Subst.	Ins.	Del.	WER
Baseline	21.9	911	364	225	14.8
Deletions only	19.2	756	325	239	15.1
Deletions + insertions	21.4	794	445	230	15.2

Table 5.6: Phone and word recognition results obtained with the tree-feature combined system, after modeling deletions only and after modeling both deletions and insertions

5.7 Detection of phonetic features in spontaneous speech

All phonetic features used in the previous experiments were extracted from TIMIT, a read speech database. For the sake of completeness, this independent section will be dedicated to the detection of phonetic features in spontaneous speech.

5.7.1 The Myosphere database

All experiments were carried out on an English telephone speech database called *Myosphere*, created by Motorola Labs. In this corpus, speakers with different dialects and foreign accents gave a set of short but spontaneous commands to an ASR system (*e.g.*, "call Steve at office"). In total, more than 100'000 commands were uttered. A small subset (about 16000 utterances)

of this database was phonetically labeled by an expert phonetician and was used for these experiments. The original inventory contained 39 phones that were a subset of the TIMIT phone inventory (cf. appendix A). However and for these experiments only, this inventory was slightly modified in order to be compatible with the phone-features conversion tables on which these experiments were based. The modifications and resulting phone lists and phonetic features can be found in appendices B.2 and B.3.

More information about this database will be given in section 7.1.1 as it will be more thoroughly used in the next chapters.

5.7.2 Phonetic feature systems

Two phonetic feature systems were evaluated:

1. The *SPE* system, based on the features given by Brondsted [17] (*i.e.*, the same system used with TIMIT). The list of features is reminded in Table 5.7 and the list of correspondences between phones and their phonetic features can be found in appendix B.2.

Feature system	Features
SPE	sonorant, syllabic, consonantal, high, back, front, low, round, anterior, coronal, voice, continuant, nasal, strident, silence

Table 5.7: The SPE system

2. The *multi-valued* (or IPA-like) system (hereafter called "MV system"), partly based on the features given by Kirshenbaum [87]. The list of feature classes and features is shown in Table 5.8 and the list of correspondences between phones and their phonetic features can be found in appendix B.3. The *nil* values in the table were used for non-relevant features; for instance in the "Voicing" class, "nil" was used to distinguish voiced consonants (marked as "voiced") from voiced vowels (marked as "nil", since voicing information is not relevant for vowels).

Feature class	Features
Voicing	voiced, voiceless, nil, silence
Place	bilabial, labio-dental, dental, alveolar, palato-alveolar, palatal, velar, labio-velar, glottal, nil, silence
Manner	stop, fricative, nasal, approximant, lateral, nil, silence
Height	high, semi-high, upper-mid, lower-mid, low, nil, silence
Front-back	front, center, back, nil, silence
Rounding	unrounded, rounded, nil, silence

Table 5.8: The multi-valued (IPA-like) system

Besides these two feature-based systems, a separate phone-based system with 34 symbols (33 phones + 1 silence) was also evaluated for comparison purposes.

5.7.3 ANN topologies

As in the previous experiments, an ANN-based system was trained for each feature system and phone set using the NICO toolkit [107]. For the SPE system, the topology was similar to the previous settings (cf. sections 4.7.3 and 5.5.3): 13 input units (12 MFCCs + 1 normalized log energy), 250 hidden units, 15 output units (14 SPE + 1 silence) and approximately 155'000 links. The phone-based system contained the same number of input and hidden units in order to obtain quite comparable results, although of course more output units (33 phones + 1 silence) and corresponding links (total: 188'000 links) were necessary. For the MV system, a separate independent ANN was trained for each feature class, with different numbers of hidden units, output units (one per feature) and connections as shown in Table 5.9. The numbers of hidden units were arbitrarily chosen, but were in similar range to those reported in the literature (*e.g.*, [85]). All ANNs included the additional characteristics described in section 4.7.3: special units for the first and second MFCC derivatives, time-delay and look-ahead connections, full connectivity between successive layers, recurrent links in the hidden layer with 50% connectivity. The number of total links in each ANN depended on these characteristics and the different attributions of hidden and output units.

Feature class	Hidden units	Output units	Connections
Voicing	50	4	21'000
Place	100	11	52'000
Manner	75	7	35'000
Height	75	7	35'000
Front-back	50	5	21'000
Rounding	50	4	21'000

Table 5.9: Number of hidden units, output units and connections in MV system-based ANNs

5.7.4 Phonetic feature targets

A difference to the experiment settings with TIMIT was that a subset of the Myosphere database was phonetically labeled, but not segmented in time. Since training and evaluation were processed on a frame-by-frame basis, time information was necessary. Therefore, a baseline HMM system was first built using the original phone inventory and the complete training set of the Myosphere database (details of the training procedure can be found in appendix C.2), then a Viterbi alignment was applied on the phonetically labeled subset to find the most likely segmentation points.

Furthermore, as mentioned previously, lists of phones used in the experiments were different from the original set in order to be compatible with the given phone-features conversion tables. All modifications were polyphonematic replacements, that is, replacements of diphthongs by two corresponding monophthongs (*e.g.*, /ow/ $\rightarrow$ /oh/ + /w/). They implied a reduction of the phone inventory (from 39 to 33 phones for both SPE and MV systems). For each replacement, instead of simply dividing the time interval of the diphthong in two equal parts, attribution of the time interval to the two monophthongs was approximated in the following way: 1) a second Viterbi alignment was applied to the same utterance, but with the diphthong replaced by its corresponding monophthongs (if a monophthong was not acoustically modeled, the closest phone model in terms of its phonetic features was used instead), 2) the relative proportions of frames returned by the Viterbi alignment for the two monophthongs were used to divide the time interval of the diphthong with the same proportions.

5.7.5 Results

Training of the ANNs followed the procedures described in the experiments of the previous chapter (cf. sections 4.7.4, 4.8.1 and King and Taylor [82]), reminded that the MV and phone-based systems had only one possible target activated at each frame, whereas more than one output unit could be activated simultaneously with the SPE system since it was an N-to-M classification. About 8000 utterances were reserved for training and the other 8000 were equally divided in two for cross-validation and evaluation purposes. For evaluation, there were a lot more silence portions in the Myosphere database (more than 60% of frames, compared to around 10% for TIMIT), which biased too much the results, so silence frames were not included in the overall statistics. Tables 5.10 and 5.11 show the results obtained for the SPE and MV systems respectively.

Concerning the SPE system (Table 5.10), performance degradation per single feature was perceptible but generally small compared to the results obtained with TIMIT in Table 5.1, and surprisingly even with some slight improvements for some of the features ("low" and "round"). However, since single feature errors did not always occur simultaneously, they had still a great impact on the "all correct" rate (= number of frames with all features correct / total number of frames), which dropped down by more than 25% relative. Some care should however be taken when comparing these results, since experiment conditions were slightly different: first, as already mentioned, silence frames were not included in the Myosphere statistics because they biased too much the results, and second, no time information was available in the Myosphere database and was estimated through Viterbi alignment, so the phone boundaries used for training and evaluation were not fully accurate. These results give nevertheless an idea of performance degradation we could expect when switching from a read speech to a spontaneous speech database.

Feature	Correct (%)	Degradation (%)	Feature	Correct (%)	Degradation (%)
sonorant	88.5	-7.2	round	94.1	+2.0
syllabic	84.1	-6.0	anterior	80.4	-10.7
consonantal	80.5	-11.5	coronal	78.4	-10.6
high	84.2	-4.3	voice	87.5	-2.1
back	83.9	-9.1	continuant	84.3	-7.8
front	83.0	-10.8	nasal	94.7	-3.1
low	93.1	+1.4	strident	92.7	-4.3
Average	**86.4**	**-6.4**	**All correct**	**39.3**	**-25.6**

Table 5.10: Frame-level classification results with SPE features on the Myosphere database, and relative degradation compared to the results with TIMIT in Table 5.1

As with the SPE system, silence frames were not included in the overall statistics of the MV system in Table 5.11 (even though results for "silence" features alone are still shown). Performance of the MV feature classes was lower than with the SPE features: their values varied from 68% for place of articulation to 76.2% for rounding, hence their average was lower by about 17.8% relative compared to the SPE system. However, these classes were multi-valued whereas SPE features were binary, and corresponding chance levels were lower as well. The results for each separate feature are much variable and reflect to a certain degree the amount of data available for training the ANNs (Chang et al. [22] also reported that the "dental" feature showed the lowest performance for the same reason). The most comparable result between the SPE and MV systems is the "All correct" rate, showing that both systems phonetically identified frames with similar performance (39.6% MV vs. 39.3% SPE).

Feature class	Correct (%)	Feature	Correct (%)	Feature	Correct (%)
Voicing	71.8	voiced	69.6	voiceless	62.9
		nil	78.6	silence	97.5
Place	68.0	bilabial	25.0	labio-dental	15.7
		dental	3.3	alveolar	69.2
		palato-alveolar	71.5	palatal	4.3
		velar	54.3	labio-velar	24.2
		glottal	18.2	nil	81.5
		silence	97.6		
Manner	68.8	stop	56.9	fricative	61.7
		nasal	58.3	approximant	47.0
		lateral	72.7	nil	80.7
		silence	97.7		
Height	68.8	high	58.8	semi-high	27.9
		upper-mid	22.4	lower-mid	58.1
		low	52.9	nil	80.1
		silence	97.1		
Front-back	72.5	front	74.2	center	31.2
		back	61.4	nil	79.9
		silence	97.0		
Rounding	76.2	unrounded	73.9	rounded	60.6
		nil	79.5	silence	97.1
Average	**71.0**	**All correct**	**39.6**		

Table 5.11: Frame-level classification results with MV features on the Myosphere database

Both systems were compared to the ANN trained directly with phone labels. The phone-based system led to a frame recognition accuracy of 55.7%, with the results per phone varying from 1.5% ([dh]) to 88.1% ([ao]), again partly influenced by the training data available. The phone-based ANN outperformed therefore both feature-based systems. However, it is reminded that:

1. Some combinations of phonetic features could not be mapped to any phone of the inventory. Forcing a valid map in the MV system to the phone with the best matching phonetic features at each frame led to a correct rate of 49.7%.

2. Feature classes and features were assumed to be completely independent. It was reported in the literature that an explicit modeling of dependencies between them could significantly improve performance (*e.g.*, classification of place of articulation features specific to each manner of articulation, see Chang et al. [22] and Wester et al. [148]).

3. Even if the best phone outputs are more accurate with a phone-based system, their alternatives (*e.g.*, 2nd, 3rd best outputs) may be phonetically farther to the reference phone than with a feature-based system (cf. experiments in section 4.8.1). So depending on the importance of alternatives (*e.g.*, creation of pronunciation networks), a feature-based system may still be a more appropriate choice.

5.8 Summary

Following the dynamic pronunciation modeling method at the lexicon level introduced in the previous chapter, we studied in this chapter a similar dynamic approach at the acoustic level based on the SLPM technique introduced by Saraçlar et al. [124]. The method can be resumed by the following three steps. First, a HMM recognizer generated a lattice of the most likely word hypotheses given an input utterance. Then, the canonical pronunciation of each hypothesis was checked by comparing a corresponding graph of phonetic features to those automatically extracted from speech using an ANN. If the comparisons showed that a phoneme of a hypothesis was likely pronounced differently, its model was transformed by sharing the Gaussian densities of its realizable phone model(s). Finally, the transformed models were combined with the original HMMs for a second recognition pass. No substantial improvement was observed on word recognition performance. However, it was shown that combination of decision trees and phonetic features that modeled substitutions and deletions led to significant reduction in phone error rate. An independent phonetic feature detection experiment on a spontaneous speech database showed a small performance degradation per single feature compared to a read speech database, but with a substantial bad impact on the overall frame level performance. A phone-based system still yielded better frame recognition accuracy than feature-based systems, but depending on some enhancements brought to the initial system and its usage (importance of alternatives), a feature-based approach may still be a more appropriate choice.

Although a dynamic pronunciation modeling approach may be appealing, one of its drawbacks is the extra processing time it requires, since such techniques need to be applied during the recognition phase. The methods proposed in the current and previous chapters are no exceptions to this fact. Furthermore, requirement of a second recognition pass represents an additional handicap. A dynamic pronunciation modeling technique incorporated into a single recognition pass would of course be preferable. A question that could also be raised about the proposed methods is the necessity of building a distinct lexicon or acoustic models per *utterance* and thus increasing the computation time, when we can reasonably assume that a speaker's pronunciation does not change much from one utterance to the next. The next chapter will propose a radically different method that will attempt to address these issues.

Chapter 6

Symbolic Speaker Adaptation: methodology

This chapter will introduce a new technique called *Symbolic Speaker Adaptation* (SSA) [98]. While this method is significantly different from the previous techniques, it overcomes several shortcomings mentioned at the end of the last chapter. The chapter is organized as follows:

- Section 6.1 will give the reasons that motivated the introduction of this new technique.
- Section 6.2 will give an insight into the SSA concept.
- Section 6.3 will compare SSA to some existing pronunciation modeling and speaker adaptation methods to see the common points and differences between them.
- Section 6.4 will present the SSA method in detail.
- Section 6.5 will give a summary of this chapter.

6.1 Introduction

In order to address the issues mentioned in the previous chapter, we are looking for a method with the following characteristics:

1. Recognition process should be done in a single pass in order to save computation time.
2. Dynamic lexicon or acoustic models should be updated based on a more general scope than a per utterance basis.

The first problem (single pass recognition) is rather difficult to solve: in order for a system to model pronunciation variations in a dynamic fashion, extra computation time is a priori needed during runtime, which implies a pre- or a post-process step. The second remark above however proposes a way to deal with the problem: it may not be necessary to build systematically a distinct lexicon or acoustic models per utterance, that is, speed issue could be addressed by considering a less dynamic system without degrading too much performance.

Given this observation, we must decide at which other level dynamic pronunciation modeling should be applied. In this chapter, we are going to consider the speaker-level, namely a

distinct lexicon or acoustic models per speaker; it is indeed reasonable to assume that transcriptions describing the pronunciations of a speaker do not radically change over time[1]. Next, we must decide how often these Speaker-Dependent (SD) lexicons or acoustic models should be updated. Let us consider the extreme cases. An update after each speaker's utterance is equivalent to the study and experiments we carried out in the previous chapters. The other situation consists of building a specific lexicon or acoustic models for a speaker who uses the system for the first time, but without modifying them during his/her future sessions. In contrast with our previous experiments and to address the speed issue mentioned earlier, we decided to study the latter case. Even though in such case the system cannot be considered fully dynamic, it still has a pseudo-dynamic behavior since its pronunciation models are modified with a change of speaker.

In such situation however, several utterances per speaker need to be analyzed in order to build reliable SD lexicons or acoustic models. This is actually a typical *speaker adaptation* problem. In speech community, the term "speaker adaptation" refers to the process that builds SD acoustic models based on more general (speaker-independent) models and some (adaptation) speech samples given by the speakers for which these SD models must be built ([152]). Since acoustic-level adaptation is already a well-known field, this chapter will only address adaptation issues at the lexical (symbolic) level: it consists for each speaker of selecting the phonetic transcriptions that best represent his/her pronunciation characteristics and of building SD lexicons from them. In order to distinguish acoustic- and symbolic-level methods in this chapter, we will call them *Acoustic Speaker Adaptation* (ASA) and *Symbolic Speaker Adaptation* (SSA) respectively. A general overview of SSA will be presented in the next section.

6.2 General overview of SSA

Let us consider a new speaker who wishes to use a speech recognition system for the first time. The system initially has no information about this new user's pronunciation characteristics, but we make the assumption that the speaker is well modeled by a blend of typical pronunciation characteristics for which the system has existing pronunciation models; let us call these characteristics *Speech Varieties* (SV) for now. The system represents this combination of SVs for each speaker by a *Speech Variety Profile* (SVP). Namely, the SVP is a definition of the speaker's pronunciation characteristics and consists of a list of speaker-associated speech varieties and their corresponding probabilities. A spatial representation of the concept is shown in Figure 6.1. A speaker is symbolized by a point in the pronunciation space, and we assume that his/her pronunciation characteristics are adequately modeled by projection of his/her representative point to a subspace spanned by a group of basis vectors. Each of these vectors can be seen as a distinct speech variety and the coordinates of the projected point according to this basis as their relative importance. A speech variety profile in this context is simply a list of these vectors and coordinates.

The objective is to identify the coordinates of the projected point, in other words to determine for the enrolled speaker the probability of each speech variety as accurately as possible. This is done through the SSA process, where the person is asked to utter a set of known (adaptation) sentences. The system builds then the SVP based on how the speaker's

[1] Actually, pronunciation can be much variable acoustically even for a single speaker (Peters and Stubley [112]). Introduction of speaker-dependent lexicons is based on two assumptions/facts: first, that inter-speaker variation is still more acute than intra-speaker variation, and second, that such variation is less visible at higher (phonetic) levels, since the number of available phones to model the acoustics is limited.

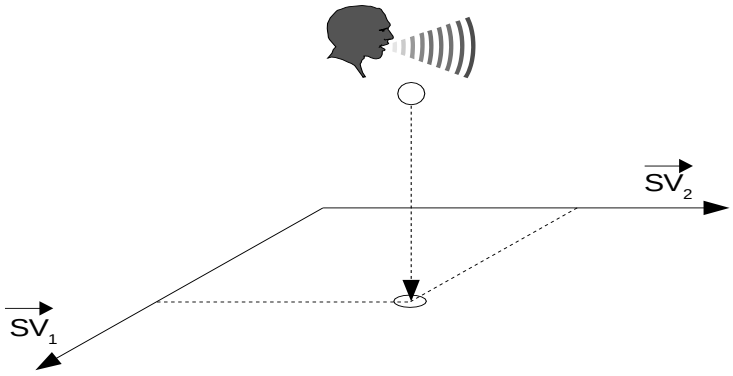

Figure 6.1: Spatial representation of Symbolic Speaker Adaptation and Speech Variety Profile

pronunciation deviates from the canonical (baseform) pronunciations.

The adapted profile is then used to expand a baseform, canonical lexicon with new pronunciation variants, the set of which is constrained by the SV probabilities contained in the speaker's SVP. Each speaker's SVP is saved for future sessions[1].

Let us now define more precisely the term *speech variety*. Each SV represents a typical pronunciation style of a given language. A logical association is to consider each SV as a dialect or a foreign-accented speech. This chapter will keep this association and will focus on speech varieties of American English. Nevertheless, this method is applicable to other languages as well.

An example of the SSA concept is shown in Figure 6.2: for a Spanish-accented English speaker, the SVP resulting from the SSA process should be biased towards the Spanish-accented English speech variety. Consequently, the SVP favors for the word "yes" the pronunciation [jh ey s] over the canonical form /y eh s/ when expanding the baseform lexicon: the "/y/ → [jh]" realization in word-initial position and "/eh/ → [ey]" are characteristics of Spanish-accented English.

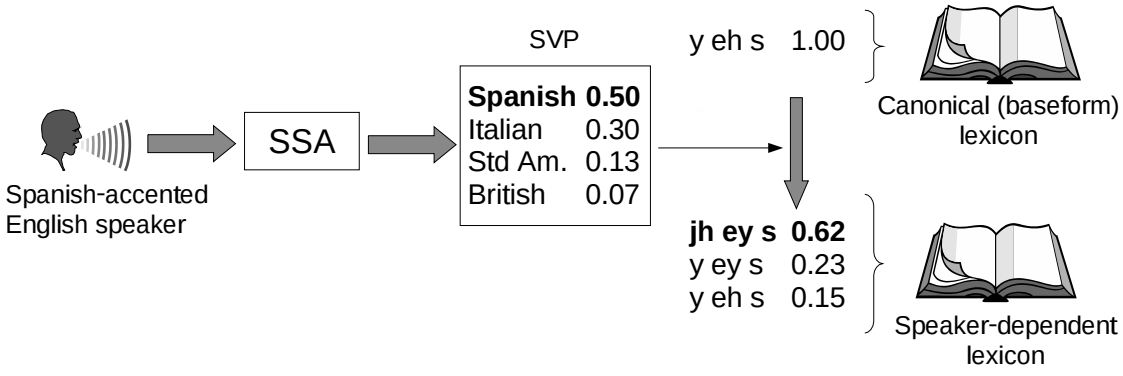

Figure 6.2: Example of SSA and SVP usage with a Spanish-accented English speaker

SSA incorporates both pronunciation modeling and speaker adaptation concepts. In order to give an insight into the potential benefits of SSA, the next section will compare it with some previous works done in both topics.

[1] A discussion of automatic speaker identification goes beyond the scope of this thesis and is not addressed here, but current existing methods in this field could be suitable for this purpose (see for example [18]).

6.3 Comparative study

6.3.1 Comparison of SSA to pronunciation modeling

As seen in the literature survey of chapter 3, most contributions in pronunciation modeling focus on generating new surface form transcriptions to better match pronunciation variability. At the same time, such methods select only the most representative variants in order to limit the risks of lexical confusability.

However, in this general strategy, pronunciation variants reflecting potentially many distinct speech varieties are inevitably omitted, which may be detrimental to ASR systems: several papers (*e.g.*, [66]) show that a speech recognizer designed for a given speech variety may sometimes double its WER when it is evaluated with a different speech variety. It would be therefore desirable to model these SVs more explicitly in order to increase pronunciation coverage.

Let us see first how this problem has been addressed so far and then compare the existing methods with SSA. Previous studies that dealt with multiple speech varieties can be divided in two categories:

1. *SV classification*, which identifies towards which SV a segment of speech is mainly biased.
2. *SV modeling*, which helps the recognition system to better match the acoustics or pronunciations of the targeted SVs.

The general method in classification consists of extracting relevant features like acoustic (*e.g.*, energy), prosodic (*e.g.*, formant frequencies) and phonetic (*e.g.*, stops, fricatives) information that help discriminate between the different speech varieties (*e.g.*, [4]), which are used as inputs to classification models such as neural networks ([15]) and LDA models ([94]). Authors of this category do not however deal with the modeling issue in general and it is difficult to see for example if and how prosodic information can also help to model pronunciation differences between speech varieties. The paper of Kat and Fung ([77]) is an exception and visualizes phoneme substitutions between two SVs by mapping the means of the first (F1) and second (F2) formant frequencies of each phoneme in the F1-F2 plane and by measuring how the points mapped in this plane are distant from each other; however, there is no way to detect any phonemic insertions or deletions with this method.

Speech variety modeling can be performed at acoustic or phonemic levels. In the former case, acoustic models can be either trained separately for each SV ([10]) - which likely requires sufficient training data - or adapted towards the targeted SV ([35]). A more interesting case for comparison with SSA is the phonemic level modeling: acoustic models of the native SV are not modified and only the lexicon is transformed, using either a knowledge-based ([77]) or a data-driven method ([3], [66], [68]). However, authors of the lexical category do not address classification issues because they typically target a single "non-standard" SV (*i.e.*, another dialect or foreign accent) or build a single joint lexicon to model several SVs.

A first benefit of SSA is that it integrates both classification (through an adaptation process) and lexical modeling. It is true that another way of targeting multiple SVs would be to combine the above classification and modeling methods in a serial fashion. However, such methods would assume that only one single speech variety is activated at a time. It would be better to have more flexibility and the freedom to activate more than one speech variety whenever needed, so that pronunciation characteristics are not represented by a single

SV-specific model, but rather a combination of them. This assumption is especially true for speakers who are best characterized by several speech varieties of the same language (*e.g.*, speakers who have lived in several dialect regions, or had parents with different speech backgrounds), but remains valid for any person who speaks with a predominate dialect or accent but sometimes pronounces words in a way better described by some other speech variety. Given this observation, it would, nonetheless, be ill-advised to merge all SV-specific dictionaries since lexical confusability increases with the number of considered speech varieties. One way to address this issue is to limit the number of pronunciations by a pruning method designed for non-native speech ([3]) and to keep a single lexicon, but in this case several pronunciation variants that best target some specific non-native speakers may still be lost. An objective and potential benefit of SSA is to keep a higher pronunciation coverage and accuracy by keeping more (if not all) variants available but by activating them at different times, namely on a SD basis through the use of speaker adapted profiles.

In brief, the potential benefits of SSA compared to the existing pronunciation models are:

- Integration of both classification and lexical modeling methods.
- Optimization towards each speaker's speech variety by the usage of a Speech Variety Profile and the possibility to combine multiple SV-specific pronunciation models.
- Higher pronunciation coverage than with pronunciation pruning methods without an excessive increase in confusability.

6.3.2 Comparison of SSA to speaker adaptation

Comparison of SSA to acoustic speaker adaptation

As mentioned in section 6.1, the objective of ASA is to build SD acoustic models from more general (speaker-independent) models and some adaptation speech data. There are basically three classes of methods applied:

1. *Direct estimation*: consists of adapting each SD model parameter separately using the available data. The two commonly used estimation criteria are Maximum Likelihood (ML), where the likelihood of the adaptation data x given the parameters λ is maximized ($P(x|\lambda)$), and Maximum A Posteriori (MAP) which additionally includes a prior distribution of the model parameters in the process ($P(x|\lambda)P(\lambda)$). The MAP criterion is preferred for adaptation purposes because less adaptation data is required thanks to the use of prior information ([96]).

2. *Linear transformation*: assumes that SD parameters can be obtained by affine transformation of the original model set. The most popular method is Maximum Likelihood Linear Regression (MLLR, [103]). In most cases, only the original means μ (*i.e.*, variances remain unchanged) are linearly modified to obtain new means $\hat{\mu}$, so that the likelihood of the adaptation data is maximized: $\hat{\mu} = A\mu + b$. The objective is to evaluate the values of A ($n \times n$ matrix, with n the mean vector size) and b ($n \times 1$ vector).

3. *Speaker clustering*: represents a speaker by a point in space, but constrains the corresponding model to be in a subspace spanned by a group of basis vectors. The objectives are first to find the most representative basis vectors and second to determine the best coordinates of the point according to this basis, in other words the weight of each vector. Essentially two methods exist in this category: Cluster Adaptive Training (CAT, [50])

and eigenvoices ([93]). CAT expresses the model mean of a particular speaker as a linear combination of "canonical" speaker clustered model means. The training scheme is iterative and consists of first finding ML estimates of interpolation weights given a set of initial clusters, then of re-estimating the cluster parameters from these weights, and so on with the updated clusters until some convergence criterion is satisfied. In the eigenvoice approach, all means of each of T SD models are gathered in a single vector to form T "supervectors", which are in turn reduced to T eigenvectors through a Principal Component Analysis (PCA); the K ($K < T$) most significant eigenvectors finally constitute the required basis. The eigenvoice coefficients (interpolation weights) are found using a maximum likelihood eigen-decomposition scheme that Kuhn et al. [93] described in their paper.

SSA is conceptually similar to the speaker clustering category since speech varieties included in speaker profiles represent a set of basis vectors. However, this procedure is still significantly different from the above methods:

- Choice of the basis is governed by pronunciation modeling since each speech variety represents a group of speakers with similar pronunciation characteristics.
- Choice of the basis is knowledge-based - the speech varieties we would like to target are known beforehand.
- SSA does *not* alter the acoustic models, but rather only modifies the speaker's SVP and the canonical lexicon, leaving the acoustic models truly speaker-independent.

ASA techniques are further divided in different *modes* of application. An adaptation process is called *static* when all adaptation data is available and is used before the actual recognition process, or *incremental* if the process is performed progressively as more adaptation data becomes available. Furthermore, ASA techniques are classified as *supervised* when the correct word sequences of adaptation utterances are known, and *unsupervised* when they are unknown. This thesis will only cover an offline mode of SSA - static and supervised. Nevertheless, it could be amenable to online (incremental and unsupervised) utilization as well - it does not face any more challenge than any other adaptation method that might be used online.

Comparison of SSA to phonetic speaker adaptation

Most contributions in speaker adaptation are applied at the acoustic level, but fewer similar experiments were carried out at phonetic level. Cohen et al. ([27]) first expanded baseform pronunciations with knowledge-based rules to create a pronunciation network for each word, then pruned these networks differently for each speaker based on SD data. An 11.5% relative improvement in WER was achieved compared to a speaker-independent baseline with multiple pronunciations. Imai et al. ([69]) derived SD phonological rules: they compared likelihoods and durations of phonemes obtained from speaker adaptation sentences (using Viterbi alignment) to some average values of the same phonemes determined from the training database. Tentative rules were generated when a phoneme had lower likelihood or duration different from the average values, and were afterwards validated if the new phoneme sequence obtained from the new rules led to a higher likelihood or a better discriminative ability between the reference and wrongly recognized sentences. The improvement they obtained was particularly significant for speakers with low baseline results (system with single pronunciation per

word), which shows that speakers with radically different accents may greatly benefit from such phonological speaker adaptation process.

SSA follows a similar line of thought, but extends the above studies by introducing the concept of speaker profiles and by explicitly taking account of multiple speech varieties. Humphries and Woodland ([68]) applied their work on SV-specific pronunciation modeling to a SD framework. Two experiments were carried out: they first built SD pronunciation models for native American English speakers from a British English recognizer using 40 adaptation sentences per speaker, then they applied the same technique to non-native speakers of English. They obtained reduction in WER with American English speakers (11% relative) and even bigger improvement with non-native speakers (19% relative), showing that phonological speaker adaptation may be particularly effective for speech varieties with substantial phonological variations compared to the native speech. This study however builds a separate pronunciation model (set of decision trees) per *speaker*, which requires *a priori* a big amount of adaptation sentences. SSA is different because pronunciation models are built per *speech variety* and adaptation sentences are only used to determine the importance of each SV-specific pronunciation model - only the lexicons resulting from the subsequent combination of models are speaker-dependent. Therefore, less adaptation data is *a priori* required.

Now that we viewed some differences and potential benefits of SSA compared to other existing methods, the following section will more thoroughly describe this method.

6.4 SSA in depth

This section will describe the SSA concept in detail. The whole adaptation process will first be explained in 6.4.1, then each component of the system will be separately presented.

6.4.1 Adaptation overview

The adaptation process is depicted in Figure 6.3. The following steps are applied for each enrolled speaker and his/her adaptation sentences:

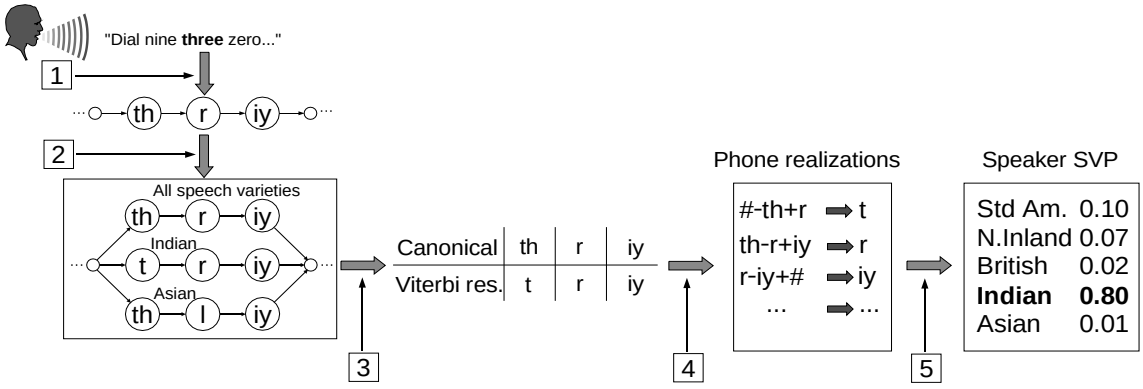

Figure 6.3: Overview of Symbolic Speaker Adaptation

1. Each word in the adaptation sentence is mapped to its baseform transcription(s) (canonical pronunciation(s)).

2. SV-specific transcriptions are derived from the baseform(s) using a pronunciation model (built in a separate process) in order to generate a pronunciation network. For each SV-specific form, a list of symbol transformations is kept.
3. A Viterbi alignment is performed using the network to return the most likely sequence of phones actually uttered by the speaker.
4. The symbol transformations corresponding to the selected phone sequence are added to a list.
5. Once all adaptation sentences are processed, probabilities for the speaker profile are computed using all transformations found in the list.

An example of a profile after adaptation for an Indian English speaker might be something like this:

Standard American English	0.10
Northern Inland English	0.07
British English	0.02
Indian English	0.80
Asian-accented English	0.01

There are four main tasks performed by the system: generation of pronunciation variants, selection of the most likely transcription uttered by the speaker, calculation of SVP probabilities and utilization of SVPs for recognition. The selection process is just a standard Viterbi alignment, so only the pronunciation models (section 6.4.2), the variant generation process (section 6.4.3), the SVP adaptation process (estimation of probabilities in SVPs, section 6.4.4) and utilization of SVPs (section 6.4.5) will be described.

6.4.2 Pronunciation models

Two different methods were investigated to expand the canonical lexicon with new pronunciation variants. The first method uses generally applicable knowledge-based rules, while the second method uses decision trees derived from the data set of these experiments.

Rules

A distinct set of rules was defined per speech variety, with each rule tagged with an *a priori* probability of being applied. Selection of rules and probabilities comes from several SV-specific studies in phonetics and phonology as well as reports and pedagogical materials concerned with English-language acquisition by speakers of other languages (*e.g.*, [20], [108], [115]). The following are some examples of SV-specific rules (formats of rules respect the description given in section 3.5.3)[2]:

Northern Inland English	/ao/ → [aa] (*e.g.*, "call" → [k aa l])
Indian English	/th/ → [t] (*e.g.*, "three" → [t r iy])
British English	/aa r #/ → [ ] (*e.g.*, "car" → [k aa])
	("#": word boundary)

[2]The complete set of rules is property of Motorola Labs and has not been published in this dissertation.

Decision trees

For each speech variety and phone combination a separate tree was trained to predict SV-specific phone(s) from a canonical phone and its left and right contexts. The following steps are applied for each training sentence:

1. On one hand, reference words are mapped using the Standard American English (SAE) lexicon to a pronunciation string. When several canonical forms are available for a given word, Viterbi alignment is applied to select the best transcription.
2. On the other hand, a pronunciation network from the baseform transcription(s) of the same reference words is generated using all sets of rules mentioned above, and then the best phone string from this network is selected using Viterbi alignment.
3. Both phone strings are aligned to each other using Dynamic Programming (DP) based on differences of their phonetic features.
4. DP alignment results are used to train decision trees. Questions used to build the trees concern phonetic features (*e.g.*, front, back, round, ...) for the immediate left and right contexts. The CART algorithm ([16]) was used to train the decision trees from the DP alignment results.

The tree building technique is similar to the version described in section 5.6.2, except that the SV-specific transcription candidate set is not obtained directly from a phone recognizer, but constrained by the application of knowledge-based rules followed by a Viterbi alignment. The reason is because a more difficult (spontaneous) database was used to carry out the experiments and phone recognizers generated too many errors (even when using phone-level bigrams)[3].

Limitations

At this initial stage of study, several constraints were observed that precluded the building of an optimal set of pronunciation models. Perhaps most crucially, only the SAE phone inventory (consisting of 39 phones) was used to describe pronunciation variability of all speech varieties. It was therefore not possible to account for non-SAE sound distinctions. For example, retroflexion of alveolar consonants is a strong acoustic cue for Indian English, but is not represented in the SAE phone inventory. This limitation prevented both the training of more specific acoustic models and the definition of additional rules to account for these sound differences (decision tree methods were also affected since their training was derived from rule productions).

Next, the *a priori* probabilities assigned to rules derive from reports of general usage in the targeted SV communities and were not re-estimated from the actual data used. Since these values help to guess the probable speech variety(ies) of the enrolled speaker (as will be shown in section 6.4.4), (likely) inaccuracies in their estimation would have (negative) repercussions throughout the system.

Finally, an overwhelming majority of sentences used to train the decision trees was uttered in SAE or in a phonologically similar SV. Although we considered the remaining sentences to

[3] An alternative method could be the use of confidence measures to filter the outputs of phone recognizers ([68])

still be sufficient for training the other speech varieties, additional non-SAE data would have been preferable in order to build more reliable pronunciation models.

6.4.3 Generation of SV-specific forms

Usage of pronunciation models occurs during two different processes: first during *adaptation* in order to create a network of possible SV-specific pronunciations per sentence, and second before *recognition* in order to output SD lexicons from the canonical (Standard American English, SAE) lexicon. Generation of pronunciation variants is almost identical in both cases. The only difference lies in the constraints applied on top of these processes:

- Creation of pronunciation network during adaptation is unconstrained (with rules) or slightly constrained (with trees): all (or almost all) pronunciations of all speech varieties are available and pronunciation probabilities are not used so that selections of SV-specific forms used to create SVPs are solely driven by the acoustics (Viterbi alignments).
- Creation of SD lexicons is constrained and governed by the enrolled speaker's SVP: only the SV-specific forms that likely reflect the speaker's pronunciation characteristics are kept and the rest is discarded. Moreover, a probability of occurrence of each form S_n given the word W it transcribes ($P(S_n|W)$) is used to reflect the relative importance of the selected forms.

This section will describe the unconstrained version. Section 6.4.4 will describe how SVPs are built and section 6.4.5 will explain how probabilities are calculated and influenced by the SVPs.

Generation of SV-specific forms using rules

In this framework, a vector of rules is associated with each speech variety. In order to transform a SAE baseform into an SV-specific form, we check the status of each rule defined for the targeted speech variety in a predefined order. Each rule is *eligible* to transform a sequence of phones only if the sequence matches the *pre-conditions* of the rule, that is, if the sequence contains the focus phoneme along with any neighbor context(s) required by the rule. Additionally, all rules are considered *optional*, which means that even if a rule is eligible, it is not necessarily applied.

The set of all SV-specific forms is obtained by all valid combinations of rules that successively transform a given baseform. Figure 6.4 illustrates the generation process through an example. Let us assume we would like to know the SV-specific forms of the word "forecast" (with baseform /f ow r k ae s t/) given a speech variety and its set of rules:

1. /# <u>th</u>/ → [t] (starting /th/ realized as [t]; '#' means a boundary)
2. /s <u>t</u> #/ → [d] (ending /t/ preceded by /s/ and realized as [d])
3. /<u>ow</u>/ → [ao] (/ow/ realized as [ao] given any phonemic context)

The first rule is not eligible since the baseform does not start with a /th/. On the other hand, the second rule is eligible because the baseform matches both the focus phoneme and

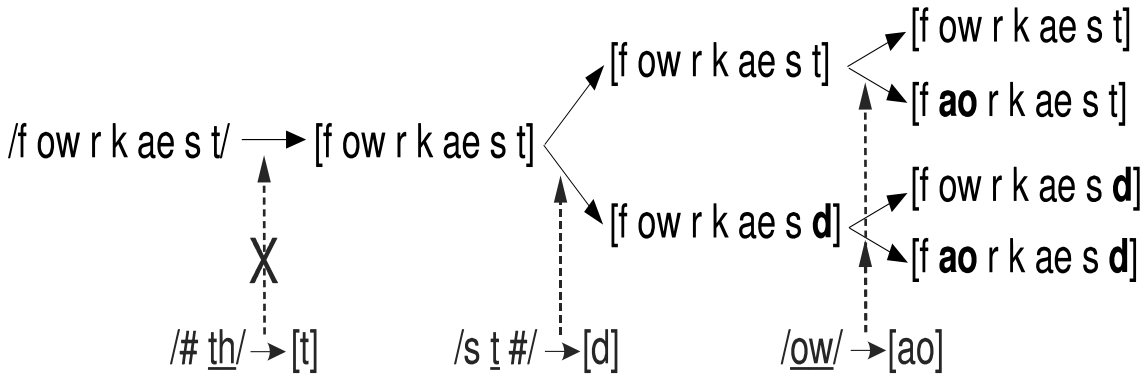

Figure 6.4: Example of generation of SV-specific forms using rules

contexts of the rule. However, the rule is only optionally fired so that the original canonical transcription is still kept. The third rule is also eligible and optional; if it is applied, all transcription outputs from the second rule (regardless of whether it has been fired or not) are used as inputs to this rule. As shown in Figure 6.4, four SV-specific forms were generated after processing the baseform through all the rules.

All rules must be applied in the order they are defined. This process is simple, but has the inconvenience of possibly causing some "feeding and bleeding" relationships between rules, that is, a rule may become eligible or on the contrary non-eligible because a previous rule in the list was fired and transformed the original transcription. For instance, if there was an additional fourth rule "/ao/ $\rightarrow$ [aa]" in the above example, it should not be eligible with respect to the baseform /f ow r k ae s t/, but may nevertheless be eligible and applicable if the third rule is applied and generates an intermediate transcription such as [f ao r k ae s t] (which contains [ao], the phone required by the fourth rule). Rules used for the SSA experiments were ordered in a way to avoid this side effect as much as possible.

Generation of SV-specific forms using trees

Given a speech variety and a baseform, phonetic features of the immediate left and right phones are used as inputs to the corresponding set of trees. The leaf reached after answering all questions found in tree branches gives the list of predicted phones along with the estimated probabilities (conditional probability of observing a predicted phone given the baseform phone and its neighbor contexts). In order to account for phone deletions and insertions as well, an output may also be a null phone (deletion) or a group of phones[4].

SV-specific forms are built from concatenation of successive outputs. Since this process generated far more forms than with rules, any predicted phone or group of phones with a probability lower than 0.1 was ignored.

[4]At this point no special control on the number of phones inserted is necessary, because the SV-specific forms used to train the decision trees are generated by rules (see section 6.4.2), in practice, that limited the number of phone insertions to one.

6.4.4 Adaptation of SVPs

The goal of SSA is to calculate the probabilities that a speaker's pronunciation characteristics match each of the speech varieties known by the recognition system. So given a speaker U who uttered some adaptation sentences $\{\sigma\}$, we compute $P(V_i|U,\{\sigma\})$ for all speech varieties V_i. On the assumption that the adaptation sentences contain sufficient information for determining a speaker's speech variety(ies), these probabilities were approximated by the sum of the contributions of all words W_j that the speaker uttered during the adaptation process:

$$P(V_i|U,\{\sigma\}) \approx \sum_{j=1}^{N(W_j)} P(V_i|W_j) \cdot P(W_j) \tag{6.1}$$

where $N(W_j)$ is the number of distinct words uttered. In related experiments (see next chapter), the importance of each word was normalized by setting $P(W_j)$ equiprobable; it did not seem indeed adequate to involve word frequencies when making a decision about a speaker's speech variety(ies), especially when some words appear much more frequently than others. For example, if a word appears 90% of the time, it is not reasonable to say that a person is an Indian English speaker only because he/she pronounces this particular word in an Indian English manner; the way how he/she pronounces the other words is just as important to make such decision.

Let us focus now on the conditional probability of a speech variety V_i given a word W_j, $P(V_i|W_j)$. In recognition of the fact that lexical words may have multiple canonical (SAE) pronunciations, $P(V_i|W_j)$ is expressed in terms of its canonical pronunciations (baseforms) B_m:

$$\begin{aligned} P(V_i|W_j) &= \frac{P(V_i,W_j)}{P(W_j)} \\ &= \frac{\sum_{m=1}^{N(B_m)} P(V_i,W_j,B_m)}{P(W_j)} \\ &= \frac{\sum_{m=1}^{N(B_m)} P(V_i|W_j,B_m)\cdot P(B_m|W_j)\cdot P(W_j)}{P(W_j)} \\ &= \sum_{m=1}^{N(B_m)} P(V_i|B_m)\cdot P(B_m|W_j) \end{aligned} \tag{6.2}$$

where $N(B_m)$ is the number of baseforms for the word W_j. Let us further develop the term $P(V_i|B_m)$ to model pronunciation variations at the canonical (baseform) level. By using the Bayes rule and simplifying the problem with the assumption that phones of a baseform are independent, we have:

$$\begin{aligned} P(V_i|B_m) &= \frac{P(B_m|V_i)\cdot P(V_i)}{P(B_m)} \\ &= \frac{\left[\prod_{b=1}^{N(p_b)} P(p_b|V_i)\right]\cdot P(V_i)}{P(B_m)} \end{aligned} \tag{6.3}$$

where p_b is a phone (in its left and right contexts) and $N(p_b)$ is the number of phones in the baseform B_m. Each phone in the baseform may be realized as: 1) itself, 2) a different phone (substitution), 3) a sequence of phones (insertion) or 4) a null phone (deletion). By summing over all possible realizations of the baseform phone p_b, we obtain:

$$\begin{aligned} P(p_b|V_i) &= \frac{P(p_b, V_i)}{P(V_i)} \\ &= \frac{\sum_{s\in\Gamma_i} P(V_i, p_b, p_s)}{P(V_i)} \\ &= \frac{\sum_{s\in\Gamma_i} P(V_i|p_b, p_s)\cdot P(p_s|p_b)\cdot P(p_b)}{P(V_i)} \end{aligned} \tag{6.4}$$

where p_s represents any phone or sequence of phones realized from the baseform phone p_b and Γ_i is the set of all valid phone realizations for the speech variety V_i (since $P(V_i|p_b, p_s) = 0$ otherwise). After substituting the expression (6.4) into (6.3) and some simplifications, we obtain:

$$\begin{aligned} P(V_i|B_m) &= \frac{\left[\prod_{b=1}^{N(p_b)} P(p_b|V_i)\right]\cdot P(V_i)}{P(B_m)} \\ &= \left[\prod_{b=1}^{N(p_b)} \frac{\sum_{s\in\Gamma_i} P(V_i|p_b, p_s)\cdot P(p_s|p_b)\cdot P(p_b)}{P(V_i)}\right]\cdot\frac{P(V_i)}{P(B_m)} \\ &= \frac{P(B_m)}{P(V_i)^{N(p_b)}}\cdot\left[\prod_{b=1}^{N(p_b)}\sum_{s\in\Gamma_i} P(V_i|p_b, p_s)\cdot P(p_s|p_b)\right]\cdot\frac{P(V_i)}{P(B_m)} \\ &= \frac{\prod_{b=1}^{N(p_b)}\sum_{s\in\Gamma_i} P(V_i|p_b, p_s)\cdot P(p_s|p_b)}{P(V_i)^{N(p_b)-1}} \end{aligned} \tag{6.5}$$

$P(p_s|p_b)$ is the speaker-dependent probability that measures how often the speaker realizes a phone p_b as p_s; it is obtained by counting the number of times this transformation occurs over all realizations of p_b during the adaptation process: $P(p_s|p_b) = \frac{N(p_s|p_b)}{N(p_b)}$. The first term of the sum, $P(V_i|p_b, p_s)$, is the SV-dependent probability and measures how accurately the same phone transformation $p_b \rightarrow p_s$ targets the speech variety V_i. Using the property of independence between p_b and V_i, and assuming that the speech varieties V_i are disjoint, we show that:

$$\begin{aligned} P(V_i|p_b, p_s) &= \frac{P(V_i, p_b, p_s)}{P(p_b, p_s)} \\ &= \frac{P(V_i, p_b, p_s)}{\sum_i^{N(V_i)} P(V_i, p_b, p_s)} \\ &= \frac{P(p_s|p_b, V_i)\cdot P(V_i|p_b)\cdot P(p_b)}{\sum_i^{N(V_i)} P(p_s|p_b, V_i)\cdot P(V_i|p_b)\cdot P(p_b)} \\ &= \frac{P(p_s|p_b, V_i)\cdot P(V_i)}{\sum_i^{N(V_i)} P(p_s|p_b, V_i)\cdot P(V_i)} \end{aligned} \tag{6.6}$$

where $N(V_i)$ is the number of speech varieties known by the system. $P(p_s|p_b, V_i)$ is given by either the *a priori* probability of the corresponding rule in the V_i rule set of being applied (if it exists, otherwise the probability equals 0) or the estimation given by the decision tree associated with the SV V_i for the $p_b \rightarrow p_s$ realization.

Finally, adaptation of speaker profiles is given by evaluating the expression seen in (6.1) with the appropriate substitutions, for each speaker and each speech variety.

6.4.5 Estimation of SV-specific form probabilities

SVPs adapted as seen in the previous subsection influence the generation of SD lexicons to be used during recognition. Namely, any speaker specific variant derived from a lexical baseform (SAE pronunciation) is assigned a probability of occurrence, which partially depends on the speaker's SVP. This subsection describes how these probabilities are obtained.

Let us consider a word W phonologically transcribed by $N(B_m)$ baseform pronunciations in the lexicon. We would like to calculate the probability of occurrence of an SV-specific pronunciation S_n given that word, $P(S_n|W)$. Since a word is entirely represented by its baseforms, we can write:

$$P(S_n|W) = \sum_{m=1}^{N(B_m)} P(S_n|B_m) \cdot P(B_m|W) \tag{6.7}$$

Pronunciation characteristics of a speaker U (who utters the word W) are represented by $N(V_i)$ speech varieties found in his/her SVP. Since several SVs may accept S_n as a possible output form derived from a baseform B_m, the probability $P(S_n|B_m)$ seen above must take all $N(V_i)$ considered speech varieties V_i into account:

$$P(S_n|B_m) = \sum_{i=1}^{N(V_i)} P(S_n|B_m, V_i) \cdot P'(V_i) \tag{6.8}$$

where $P'(V_i) = P(V_i|U, \{\sigma\})$ is the probability that the speech of the speaker U conforms to the i-th speech variety (see section 6.4.4).

The process to evaluate $P(S_n|B_m, V_i)$ differs between the decision tree and rule methods. These two processes are explained next.

SV-specific form probabilities using trees

Evaluation of $P(S_n|B_m, V_i)$ using decision trees is quite straightforward. Each phone p_b of the baseform B_m is realized as p_s that represents the same phone p_b, a distinct phone (substitution, deletion) or a group of phones (insertion). The SV-specific form S_n is obtained by simply concatenating the successive p_s realized from each baseform phone p_b. Assuming that the p_b's are independent, we can write:

$$P(S_n|B_m, V_i) = \prod_{b=1}^{N(p_b)} P(p_s|p_b, V_i) \tag{6.9}$$

where $N(p_b)$ is the number of phones in the baseform B_m. Each term of the product is estimated by the decision tree associated with the speech variety V_i and baseform phone p_b.

SV-specific form probabilities using rules

Let us focus on the rules responsible for the transformation of a baseform B_m to an output form sequence S_n, to see how they influence the probability $P(S_n|B_m, V_i)$. In this framework, each speech variety V_i is associated with a vector (ordered set) of rules $r_i = (r_i^1, r_i^2, ...)$. As mentioned in section 6.4.3, each rule r_i^j of the set may be eligible if the required pre-conditions of the rule are met, but even so it is not necessarily applied. To represent these possible rule states in a more compact form, we define a variable $q_i = (q_i^1, q_i^2, ...)$, where each q_i^j represents the state of a rule r_i^j, with three possible values:

1. '0': the rule is not eligible
2. '+': the rule is eligible and applied
3. '-': the rule is eligible, but not applied

We come back now to the process of transformation of a baseform B_m to an SV-specific form S_n, but this time bringing the rules and rule states to the fore. To find the probability $P(S_n|B_m, V_i)$ of equation (6.8), we are looking for all combinations of rules that successively transform the baseform B_m into the SV-specific form S_n: $B_m \rightarrow X_1 \rightarrow X_2 \rightarrow ... \rightarrow S_n$. Conditioned to a speech variety V_i and its set of rules R_i, it consists of finding those sequences of rule states q_i for the rule set r_i that leads to S_n. Therefore, the probability becomes:

$$P(S_n|B_m, V_i) = \sum_{q_i \in Q_i} P(B_m \stackrel{q_i}{\rightarrow} S_n) \tag{6.10}$$

where Q_i is the set of valid sequences of rule states that transform the baseform B_m to the SV-specific form S_n for the given speech variety V_i, provided that at least one such sequence exists (otherwise the probability becomes zero). If the set Q_i is not empty, each term of the sum in equation (6.10) can be expressed as a product of probabilities of rules being in the required state to yield the output form S_n:

$$P(B_m \stackrel{q_i}{\rightarrow} S_n) = \prod_{j=1}^{L_i} P(q_i^j) \quad \text{iff } B_m \stackrel{q_i}{\rightarrow} S_n \tag{6.11}$$

where L_i is the number of rules defined for the speech variety V_i. Furthermore, each rule state probability can be expressed as a function of the *a priori* rule probabilities (defined from knowledge-based sources):

$$P(q_i^j) = \begin{cases} 1 & \text{if } q_i^j = \text{state '0'} \\ P(r_i^j) & \text{if } q_i^j = \text{state '+'} \\ 1 - P(r_i^j) & \text{if } q_i^j = \text{state '-'} \end{cases} \tag{6.12}$$

Finally, the probability associated to each selected SV-specific form S_n for a word W is the value of $P(S_n|W)$ with the appropriate substitutions seen above.

6.5 Summary

In this chapter, we introduced a method called *Symbolic Speaker Adaptation* (SSA) in order to overcome several disadvantages seen with the methods based on phonetic features. The method consists in building speaker-dependent lexicons based on an adaptation process that help to capture the most relevant symbol transformations realized by each speaker. It incorporates both speech variety classification and modeling tasks, and is different from Acoustic Speaker Adaptation (ASA) in that only the lexicon is altered, leaving the acoustic models speaker-independent.

The next chapter will describe the related experiments and results.

Chapter 7

Symbolic Speaker Adaptation: experiments and results

This chapter will describe all experiments and results obtained by applying the SSA method seen in the previous chapter [98][99]. Sections are organized as follows:

- Section 7.1 will describe the basic settings (database, HMMs, ...) and experiments carried out.
- Section 7.2 will analyze the results obtained with the basic experiments.
- Sections 7.3, 7.4 and 7.5 will present some additional experiments pertaining to the analyses and hypotheses made in section 7.2, in order to further improve recognition performance.
- Section 7.6 will evaluate the robustness of SSA under some constraining situations (small adaptation data, non-modeled speech varieties).
- Section 7.7 will give a summary of this chapter.

7.1 Basic experiments

7.1.1 Database

All experiments were carried out on an English telephone speech database called *Myosphere*, developed by Motorola Labs. In this corpus, speakers from 12 speech varieties give a set of commands to a real speech recognizer (*e.g.*, "call Steve at office"). Most commands are short (3.8 words per sentence on average), but spontaneous and often uttered with hesitations and in different noisy conditions (*e.g.*, cross-talk, line noise), so they represent fairly well a real life situation. The original phone inventory contains 39 symbols, similar to those used in TIMIT (cf. appendix A). Speech files include several annotations, including the speaker gender and his/her dominant speech variety. The represented speech varieties are:

1. Standard American English
2. Northern Inland English (*e.g.*, Chicago)
3. Southern English

4. African-American English
5. New York English
6. British English
7. Indian English
8. Asian-accented English
9. Spanish-accented English
10. French-accented English
11. German-accented English
12. Other (unknown speech variety)

However, distribution of sentences is biased towards Standard American English and Northern Inland English (a dialect close to Standard American English): around 80% of sentences were uttered by speakers of these two speech varieties.

7.1.2 Baseline system

A baseline HMM system was trained using HTK [158]. Around 90000 sentences uttered by about 440 speakers were used for training. All 12 speech varieties were included (although biased towards SAE and Northern Inland English, as mentioned previously). Models consist of 39 monophones with 5 Gaussian mixtures per state, trained from 39 MFCC coefficients (12 static + 1 energy, 13 Δ, 13 $\Delta\Delta$). Two additional models for silence and short pause were also trained, giving a total of 41 models. As mentioned in section 6.4.2, no models specific to non-SAE SVs were used. Details of the training process can be found in appendix C.2.

Five speech varieties were used for evaluation: Standard American English (SAE), Northern Inland English (NI), British English (Br), Indian English (In) and Asian-accented English (As). Eight to ten speakers (4-5 male, and 4-5 female) with an average of 164 sentences per speaker were used for each speech variety evaluation. A back-off bigram language model which was generated from all sentences of the database helped constrain the search[1]. Two different baseline lexicons created by Motorola Labs were used: the first lexicon (BLex1) contained only one baseform pronunciation per word, while the second lexicon (BLex2) was an expanded version of the first one with pronunciation variants created by phoneticians[2]. The closed vocabulary size for both lexicons was 3815 words. Table 7.1 gives the baseline recognition results in WER.

	SAE	NI	Br	In	As
Base (BLex1)	18.92	21.60	36.95	24.37	32.92
Base (BLex2)	18.31	20.92	34.93	23.45	31.31

Table 7.1: Baseline recognition results (in WER)

The table shows that:

- As expected, the expanded lexicon (BLex2) leads to a lower WER than the basic version (BLex1) for all SVs, although not by much (between 3.1% and 5.5% relative improvement).

[1] Test sentences were voluntarily included so that the out-of-vocabulary problem would not influence the results of our experiments.

[2] The average number of pronunciations per word in BLex2 is slightly higher than the CMU [26] and BEEP [11] dictionaries.

- The WER may substantially increase with the speech variety considered (*e.g.*, the Br WER is almost double the SAE WER). It is surprising to see that performance is the worst with British English speakers when we know that English is their very native language. Observation of results on a per speaker basis shows that WER was even around 50% for some of the Br speakers. It is interesting to note that Humphries [67] also reported high WER for one of the two British English speakers he evaluated with SAE trained acoustic models, compared to WERs of non-native speakers.

Results obtained with BLex2 were chosen as baseline results for the SSA experiments reported in the next subsections.

7.1.3 Training of decision trees

A program (called *wagon*) from the speech tools of Edinburgh University [143] was used to train the decision trees. As explained in section 6.4.2, a separate tree was trained for each speech variety and phone. Sets of rules were applied to the BLex2 lexicon to generate candidate transformations. Pronunciation variants found in BLex2 were also transformed in this process because they reflected pronunciation variability implied by connected speech and not by dialectal variations (like in "Barbara": /b aa r b ah r ah/ $\rightarrow$ [b aa r b r ah]), so could still be modified by SV-specific rules.

Please note that due to a lack of data, trees for the As SV had to be trained from the test set as well, and therefore all tree-related results for the As SV reported in the next sections are for indication only. However, the SSA process itself strictly uses the adaptation set for all speech varieties. In other terms, the As case gives an idea how effective the SSA method would be if the pronunciation models were "perfect" (*i.e.*, if the pronunciation models well matched the evaluation data).

7.1.4 Results with SSA

The SSA method was applied to the whole adaptation set (140 sentences on average per speaker[3]). Each network created for Viterbi alignments had on average 7.5 pronunciation variants per word with rules and 10 with decision trees. Before computing the SVP probabilities, SV-specific phone realizations that occurred less than 5 times were pruned to keep only reliable transformations. All speech varieties, words and baseforms used to compute the SVP probabilities were considered equiprobable ($P(V_i) = 1/N(V_i)$, $P(W_j) = 1/N(W_j)$ and $P(B_m) = 1/N(B_m)$) so that the final results were not biased towards any speech variety without any knowledge about the speaker's pronunciation characteristics. Some additional pruning mechanisms – maximum 3 pronunciations per word, and stop when the sum of the highest output form probabilities equals or exceeds 0.7 – were also applied to the generated user lexicons to keep the lexicons small and to limit lexical confusability. Probabilities of pronunciation variants were also scaled during decoding to emphasize their importance relative to acoustic and language model scores. Table 7.2 shows the results obtained.

Implementation of the SVP concept at this stage showed very little improvement relative to the BLex2 baseline results. Nevertheless, the following points can be noted:

[3]This seems a large dataset, but since sentences are short they are equivalent to 30-35 sentences of Wall Street Journal (WSJ0) in terms of number of words.

	SAE	NI	Br	In	As
Base (BLex2)	18.31	20.92	34.93	23.45	31.31
SVP rules	18.85	21.05	35.72	23.37	31.40
SVP trees	17.99	20.86	34.36	23.85	29.36

Table 7.2: First results with SSA (in WER)

- Decision trees are in general more effective than purely knowledge-based rules. This is first because each SV-specific tree was trained based on *all* available SV-specific rules (and not only the corresponding SV-specific rule set). This method offered more flexibility by allowing rules to be shared across speech varieties (the rule method did not allow it, unless a same rule was explicitly mentioned in each rule set). Secondly, symbol transformation probabilities were data-driven, in comparison with the rule method where probabilities were defined *a priori* and were not re-estimated from the data.
- The As SV (case when pronunciation models well match the evaluation data) with decision trees still shows some small improvement (6.2% relative in WER).

Since results were rather disappointing, the following cheating experiment was run to get an idea about the maximum improvement we can expect from SSA under current conditions: since each speaker in the database was tagged with his/her dominant speech variety, the SVP for each speaker was completely biased towards it before recognition. For example, if a person was known to be an Indian English speaker, an SVP with a probability 1.0 for Indian English and 0.0 for the other SVs was created for him/her and this SVP was used for recognition instead of an SVP that would have normally been generated from the adaptation process. Table 7.3 shows the results obtained.

	SAE	NI	Br	In	As
Base (BLex2)	18.31	20.92	34.93	23.45	31.31
SVP rules (cheat)	18.31	20.92	35.41	24.65	32.49
SVP trees (cheat)	17.35	20.53	34.68	23.66	28.74

Table 7.3: Cheating experiment results with SSA (in WER)

Surprisingly, even the cheating experiment did not bring any substantial improvement either. Decision trees still perform better than rules, but even so results are not much better – if not worse – than the basic SSA experiment results reported in Table 7.2. Section 7.2 will try to understand the reasons.

7.1.5 Comparison with acoustic speaker adaptation (ASA)

The same experiments as mentioned in the previous subsection were carried out on ASA-adapted HMMs. The ASA method used was Maximum Likelihood Linear Regression (MLLR) with an 8-base regression class tree to cluster acoustically similar mixture components before evaluating the transformations (see [49] or [158] for more details). The amount of adaptation data was the same as for the SSA technique. Table 7.4 shows that application of ASA is much more effective than SSA. However, we also notice that SSA performs better when combined with the ASA technique (up to +7.8% relative improvement with decision trees over the ASA baseline, +11.7% for the As SV). It seems that since lexicons generated by SSA are speaker-dependent, they work better when the acoustic models are also speaker-dependent. SSA and

ASA are applied at distinct levels of the system, and these results suggest that they are complimentary.

	SAE	NI	Br	In	As
ASA+BLex2	11.74	12.96	20.59	13.91	19.04
ASA+SVP rules	12.13	12.82	20.53	14.18	19.18
ASA+SVP trees	11.04	12.48	18.99	13.26	16.81

Table 7.4: Results of SSA techniques over ASA (in WER)

It is interesting to note that Willett et al. [150] carried out some similar but independent experiments and also observed the same kind of result. Their method consists for each speaker in running a first word recognition pass, then in applying some knowledge-based rules on the resulting sequence of phoneme models in order to generate a pronunciation network. A Viterbi alignment through the network selects the best model sequence and helps to weight the rules involved for the specific speaker. Based on that, a speaker-dependent lexicon with multiple pronunciations and probabilities is generated and is used instead of the original (speaker-independent) lexicon for a second word recognition pass. Although improvement on their Japanese database was small, they noticed that joint acoustic (using MLLR) and symbolic adaptation applied between the first and second recognition pass were complimentary and that the corresponding improvements were additive. Recently, Goronzy [55] (chapter 8) generated non-native pronunciation variants from the data of source and target languages: some German speech data was transcribed using an English trained phone recognizer to obtain an estimation of English-accented pronunciation variations, and the correspondences between canonical pronunciation and accented variants were captured through a decision tree. Experiments showed that combination of MLLR and a lexicon enhanced with non-native variants generated with the tree improved performance compared to MLLR alone, suggesting once again that acoustic and pronunciation levels are additive (at least partially). Amdal [2] (chapter 7) on the other hand obtained only small or no improvement by combining pronunciation variants and acoustic adaptation. It should however be noted that the variants used were of general purpose (taken from the CMU lexicon) and already degraded performance without speaker adaptation on spontaneous and non-native speech tasks.

7.2 Result analysis

In order to understand why the SSA experiments failed to bring more substantial improvement, results of Viterbi alignments were analyzed on the adaptation data. They show that:

1. Many pronunciations selected by the Viterbi alignments were associated with more than one speech variety: 78% with rules and 87% with trees (77% with rules and 64% with trees were common to all 5 SVs).
2. Many selected pronunciations were baseforms found in the BLex1 lexicon: 71% with rules and 54% with trees.

Given the first remark, the more speech varieties which accept a selected pronunciation as a possible SV-specific form, the more difficult it is to decide which speech variety best describes a speaker's pronunciation, especially when the amount of preference for a selected transcription is similar across SVs. The following lines give an example of Viterbi alignment result for the word "change" when using rules :

Selected pronunciation : [ch ey n jh]
Winners : sae(1.00) ni(1.00) rp(0.93) in(0.80) as(0.81)

The value next to each speech variety is the probability that a speaker realizes the baseform of the word "change" as the form [ch ey n jh] (in this example, phonetic transcription matches the baseform) assuming that his/her dominant speech variety is each of the considered SVs. As seen above, not only all SVs are possible, but also none of them is much more preferred than another. When such case happens frequently (as it is the case), final SVP probabilities are influenced in the same way.

The second remark tells us that since baseforms are often preferred, the SVPs should be biased towards those speech varieties that most resemble SAE, namely SAE and NI. It is indeed the case with the knowledge-based method (rules). Figure 7.1 shows the average SVP probabilities obtained for each of the modeled SVs (for example, the five leftmost columns show the average SVP values for SAE speakers). It shows that SAE and NI (sometimes Br[4]) have the highest values for all speakers, although not by much as noted in the first observation above (all SVP probabilities range between 0.15 and 0.24).

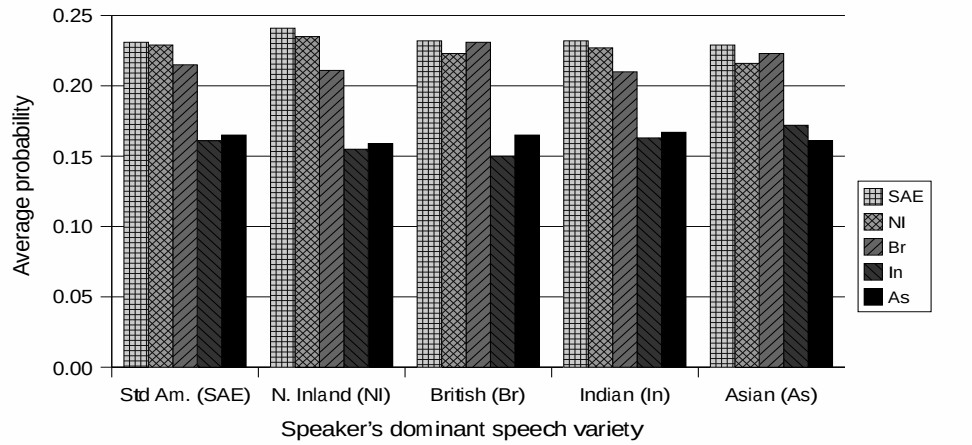

Figure 7.1: Average SVP probabilities using rules for each modeled SV

Decision trees (Figure 7.2) do not completely follow this assumption since preference for baseforms by a given SV is data-driven. They bias the SVPs slightly more towards the targeted SVs, but again the preference for one SV over another is not great.

Since baseforms are often preferred, SV-specific variants added to the lexicon can increase lexical confusability more than they help with modeling pronunciation variation, which may explain the lack of substantial improvement.

Following these remarks, a question we could ask ourselves is: why this preference for baseforms ? Some possible answers are:

1. Speakers really pronounce words in a rather canonical fashion.

2. Something lacks and prevents the ASR system from modeling pronunciation variations more accurately.

[4]Compared to In and As, fewer pronunciation variants distinguished Br from SAE and NI.

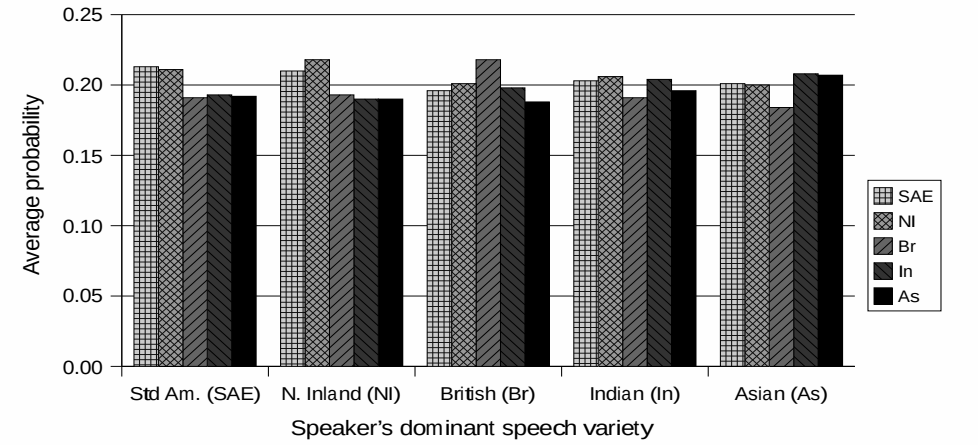

Figure 7.2: Average SVP probabilities using trees for each modeled SV

Let us consider the first hypothesis. It is true that speakers in this database were knowingly interacting with an ASR system and voluntarily spoke carefully so that their requests could be understood. It is possible that pronunciations resulting from hyperarticulations are often best described by baseform pronunciations. One way to verify it would be to run the same experiments on another database in which speakers of different speech varieties spoke more naturally, and check if preference for baseforms is reduced in favor of SV-specific pronunciation variants. However, at the time the experiments were carried out, a spontaneous speech database with multiple speech varieties and besides annotated with the dominant SV for each speaker was not easy to find, so such experiment will not be reported in this thesis. But regardless of the possible results of this experiment, it is obvious when listening to various non-SAE speakers that pronunciations across speech varieties are fundamentally different – otherwise it would be difficult to reliably annotate the dominant SV of each speaker in the first place.

This naturally leads to the second hypothesis (lack of accuracy of the ASR system). If a baseform is often selected by Viterbi alignments even if it is not the true pronunciation, it is because:

1. Acoustic models are not accurate enough to choose the true pronunciation.
2. The true pronunciation is not among the possible choices and the baseform is chosen for lack of better phonetic transcription.

Let us study each of the reasons (and we will recall some of the limitations already seen in section 6.4.2). Concerning the first reason, the acoustic models used for the reported experiments were monophones, so maybe more accurate models (like triphones) could help to decrease the amount of selected baseforms in favor of other pronunciation variants (assuming they better reflect true pronunciations than the baseforms). Another possible explanation is due to the database used to train the models: since the database was heavily biased towards the SAE and NI speech varieties (which both favor baseforms), it is possible that Viterbi alignment results were consequently biased in the same way. A training with a database equally balanced in SVs may help to decrease the amount of baseforms selected.

If the second reason (*i.e.*, true pronunciation not present) is correct, current pronunciation models need to be improved to generate the right transcriptions. We need then to study why the generation process is not accurate enough. A lack of pronunciation rules could be

one of the factors. However, we recall that current rules and decision trees already made available a reasonable number of transcriptions per word (7.5 transcriptions with rules on average and 10 with decision trees) and since they are based on linguistic knowledge, they are supposedly realistic; despite of this fact, Viterbi alignment decisions still gave baseforms as winners in many cases. It is likely that just adding more rules will not help much. A better explanation of the problem seems to be associated with a lack of non-SAE phones: non-SAE speakers pronounce fairly differently from SAE speakers, but since typical new sounds (*e.g.*, retroflexion of /t/ in Indian English) have not been taken into account, an alternative phonetic transcription is chosen for lack of better one. Since baseforms are the most representative transcriptions across SVs, they are generally preferred over the rest.

We propose to further study the various hypotheses mentioned in the next sections (some of them were reported in [99]). To summarize, the following ASR components have been pointed out for an update and their effects will be analyzed:

- Acoustic models - triphones (section 7.3)
- Acoustic models - training data (section 7.4)
- SV-inclusive phone inventory (section 7.5)

7.3 Experiments with triphones

5770 word-internal triphones were built by cloning the single Gaussian version of the 39 monophones used in previous experiments and by re-estimating their parameters using the Baum-Welch algorithm and triphone-labeled transcriptions. Then, the triphones were clustered with a data-driven process, which reduced the number of models to 3673. Finally, the number of mixtures was progressively increased to five, with four Baum-Welch re-estimations before each mixture increase. The same SSA experiments as with monophones were carried out with triphones. New decision trees were trained for triphone models following the same settings as reported in sections 6.4.2 and 7.1.3. Table 7.5 compares between monophones and triphones the percentage of selected pronunciations shared between SVs and preference for baseforms when using decision trees. We see that the percentages are even higher with triphones.

	Shared prons	Baseforms
Monophones	87%	54%
Triphones	93%	79%

Table 7.5: Percentage of selected pronunciations shared between SVs and preference for baseforms with monophones vs. triphones

Since preference for baseforms is even more accentuated, even less relative improvement than with monophones was predicted with triphones. Table 7.6 confirms this expectation. Decision trees were not at all effective with triphones because too few pronunciation variations were observed: symbol transformations (in broad sense, that is substitutions, insertions and deletions) occurred only 3.73% of the time in the training data (vs. 12.94% with monophones), so that 77% of the resulting trees contained only a single node.

	SAE	NI	Br	In	As
Base (BLex2)	11.69	12.49	23.07	20.66	20.72
SVP rules	11.70	12.84	23.26	20.56	20.90
SVP trees	11.55	12.30	23.02	20.75	20.82

Table 7.6: Comparison of baseline and SSA performance with triphones (in WER)

7.4 Influence of an SV-balanced training

In order to see to what extent availability of non-SAE training data influences the preference for baseforms and recognition performance, two new different sets of HMMs were trained. The first set (*SAE-only*) was trained using 14016 sentences of SAE data only, while the second set (*Multi-SV*) was trained using 14016 sentences evenly balanced (3504 sentences each) between Standard American English (SAE), Northern Inland English (NI), British English (Br) and Indian English (In)[5]. The same training process as in section 7.1.2 was applied to build the models, except that the number of Gaussian mixtures was increased to ten. Sentences used for evaluation were uttered by nine to ten speakers of each of these four SVs.

Table 7.7 compares between SAE-only and Multi-SV HMMs the percentage of shared pronunciations and preference for baseforms when modeling with decision trees. We notice that Multi-SV HMMs have similar behavior to triphones: as acoustic models become more accurate, preference for baseforms increases along with the percentage of shared pronunciations.

	Shared prons	Baseforms
SAE-only	85%	53%
Multi-SV	92%	68%

Table 7.7: Percentage of selected pronunciations shared between SVs and preference for baseforms with single SV (SAE) training vs. Multi-SV training

Next, Table 7.8 shows the recognition results. It is not surprising that the Multi-SV HMMs outperform the SAE-only HMMs with speech varieties significantly distinct from SAE, namely Br and In. Also, as would be expected, using models trained with 75% of non-SAE data (Multi-SV) rather than 100% SAE data (SAE-only) causes the WER for SAE test data to rise, but only moderately. The last line shows the performance brought by application of SSA on the Multi-SV HMMs. Since baseforms still constitute the majority of selected pronunciations, again no particular improvement could be observed.

	SAE	NI	Br	In
SAE-only	17.12	19.21	36.65	26.18
Multi-SV	17.97	19.35	25.68	21.94
Multi-SV + SSA	17.77	18.92	24.73	21.89

Table 7.8: Recognition results (WER) with single SV (SAE) training vs. Multi-SV training without and with SSA

[5]Asian-accented English could not be included because too few training sentences were available.

7.5 Influence of an SV-inclusive phone inventory

According to the results in previous sections, preference for baseforms is not because models are not accurate enough to target true pronunciations, but rather because true pronunciations are not among the available choices. As previously suggested, a way to better incorporate true pronunciations when multiple SVs are involved is by increasing the phone inventory size in order to model non-SAE sounds. The following experiment was therefore set: instead of training a single acoustic model per phoneme, two or more models were built to take account of SV-specific phones referring to the same phoneme (like the way we would do with triphones to take account of different phonetic contexts). For this purpose, four additional sets of HMMs besides the initial 41 symbol set were trained, with 70, 100, 130 and 164 symbols respectively. The HMM set with 164 symbols was obtained by training four subsets of 41 SV-specific models using each corresponding subset of 3504 SV-specific training sentences from section 7.4. The symbols (appropriately tagged for SV) were then simply combined at the end of training. The remaining sets (70, 100 and 130) were trained like the 164-set at the initial stage, but their HMM states were then clustered with 3 different threshold levels (yielding 70, 100 and 130 models) before the number of Gaussian mixtures in each state was increased. Separate pronunciation models using decision trees were also built for each set of HMMs. Training process was similar to the method described in section 6.4.2, except that, in order to take account of the new phones introduced, each original phonetic transcription found in pronunciation networks created during the training of decision trees had four versions, each referring to one of the four subsets of SV-specific phones.

Like in the previous experiments, the percentage of shared pronunciations and preference for baseforms were measured and reported in Table 7.9 for each phone inventory size. In contrast to the previous results, explicit modeling of non-SAE sounds helped to substantially decrease both values (down to 16% of shared pronunciations and 14% of baseforms for the biggest inventory set). A closer look showed that the percentage of pronunciations shared by *all* SVs dropped the most (less than 1% remaining with 130 and 164 symbols).

	Shared prons	Baseforms
41 symbols	92%	68%
70 symbols	85%	50%
100 symbols	72%	36%
130 symbols	50%	26%
164 symbols	16%	14%

Table 7.9: Percentage of selected pronunciations shared between SVs and preference for baseforms with different sizes of phone inventory

Since phone inventory expansion was able to reduce the preference for baseforms, we could reasonably expect that SSA would better be able to match each targeted speech variety (instead of biasing the results towards SAE and NI). Figure 7.3 gives the average SVP probabilities of each modeled SV using decision trees as pronunciation model and the 164 symbol inventory. In contrast to the results shown in Figure 7.2, this adaptation method was better able to hone in on speakers' speech varieties, especially with British English speakers. As previously mentioned, SAE and NI (*e.g.* Chicago) speakers have very similar pronunciation styles and are easily confusable each other, which explains the lower probabilities obtained; merging the two SVs would yield a probability close to the value obtained for Br. The Indian SV is on the other hand more surprising, since its average probability is still biased towards SAE and NI. Nevertheless, the "In" column for Indian speakers is the highest among the other

"In" columns shown in the figure. We will come back to the issue with Indian SV at the end of this section.

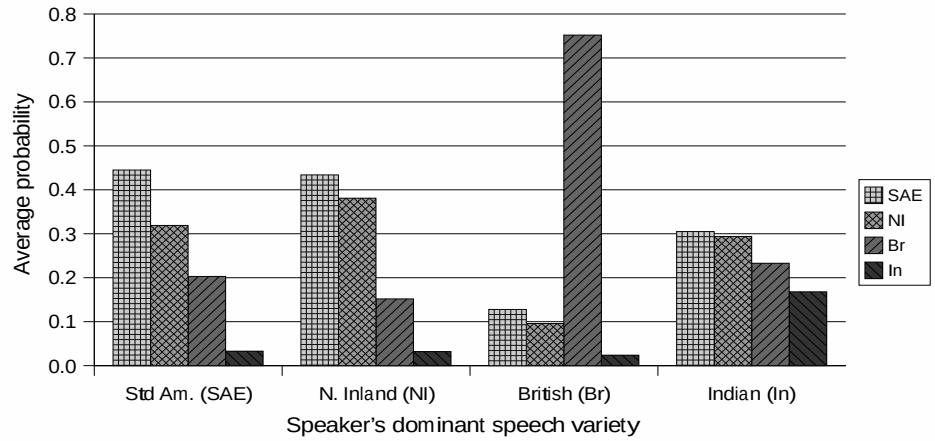

Figure 7.3: Average SVP probabilities for each modeled SV using 164 symbols

Each set of models was then evaluated for baseline performance. At this stage, a decision had to be made about the recognition lexicon: should it include the non-SAE symbols, and if so, how should they be included ? Taking account of new symbols seemed *a priori* logical, but its incorporation was not as easy: the more symbols were included and the more phonetic transcriptions per word were added to the lexicon, which increased lexical confusability. Combination of symbols from distinct SVs to build even more pronunciation variants made of course the situation worse. An optimal selection of phonetic transcriptions per word was not straightforward. Moreover, it seemed more logical to keep as many components as possible identical across the different sets of HMMs to make a fair comparison. So finally, it was decided to keep the same recognition lexicon for all models and focus on the effects of increasing the phone inventory without modifying anything else.

	SAE	NI	Br	In
Base 41	17.97	19.35	25.68	21.94
Base 70	17.41	19.23	26.47	21.95
Base 100	16.59	19.54	29.27	24.32
Base 130	16.57	18.78	33.52	26.26
Base 164	17.22	19.67	35.27	26.44
Base cheat	17.22	18.59	23.39	24.40

Table 7.10: Baseline results (WER) with SV-inclusive (expanded) phone inventories

Table 7.10 shows the baseline WER test results for each of the included SVs (columns) and for each of the trained HMM sets (rows). To get an idea of the expected upper bound on performance, a "cheating experiment" was also run for each SV, recognizing the test utterances for each SV with the 41 models trained exclusively on the 3504 training utterances for that SV. Those results are included as the last line in Table 7.10, labeled as "Base cheat" results. Given the choice of keeping the SAE lexicon across all HMM sets for recognition, SVs noticeably different from SAE (that is, Br and In) show a significant performance degradation when phone inventory is expanded, because the SAE models used for recognition were trained with less non-SAE data. It should be noted that for the maximum inventory size (Base 164), the SAE models were trained solely from SAE data and results are therefore similar to the "SAE-only" results reported in Table 7.8 (remaining differences reside in the amount of training

sentences, 3504 in this case vs. four times this amount in the previous case). Moreover, again due to the use of SAE symbols only for recognition, the Base 164 and cheating experiments for SAE become one and the same and therefore end up with the same result (17.22% WER). Finally, we surprisingly notice that for the Indian SV, cross-SV collapsed models yield even better performance (10% lower WER) than specific Indian SV models used for the cheating experiment.

Examination of the SSA WER results in Table 7.11 leads us to note that, with the exception of a few cases (mainly within SAE which was already well matched in the baseline), the SSA process consistently leads to WER reductions relative to the corresponding baseline (non-SSA) results. We also note that for all SVs except Indian, there is at least one expanded inventory that produced better WER as that yielded by the SSA process with the original minimal phone set (SSA 41). Compared to the Base 41 baseline, the improvement was best for the Br SV with 9.9% relative reduction in WER (overall WER reduction over the SAE-only trained HMMs was 36.9%). Unfortunately, the relationship between phone set size and best WER is not consistent across SVs and therefore rather opaque. Once again, results for the In SV are rather puzzling. Though SSA does generally yield WER reductions for In relative to the corresponding non-SSA results, they are not nearly as dramatic as those for Br. And, rather than following the pattern of improving SSA results with larger sets of (more precise) models to recruit from, SSA performance decreases with model inventory size for In.

	SAE	NI	Br	In
SSA 41	17.77	18.92	24.73	21.89
SSA 70	17.60	18.78	24.73	21.91
SSA 100	16.27	19.26	24.53	22.84
SSA 130	17.70	18.64	23.89	25.83
SSA 164	17.70	19.00	23.13	26.73

Table 7.11: SSA results (WER) with SV-inclusive (expanded) phone inventories

Results observed in this section for Indian English (In) remain somewhat of an enigma. First, it is strange that SSA did not target the Indian SV for Indian English speakers. Second, larger gains in performance were obtained with multi-SV trained models than with models trained exclusively on Indian English data. Some Viterbi alignment analyses showed that even though Indian English trained models are acoustically more accurate on Indian English *training* data than SAE models (higher average acoustic likelihood per phone), they are less accurate with Indian English *adaptation* data. Consequently, nearly twice as many selected pronunciations during the SSA process were in favor of SAE (39% of selected pronunciations for SAE vs. 21% for In), hence the bias of Indian English speakers' SVPs towards SAE. A possible explanation of the problem could be associated with the training data: due to a lack of availability, only a limited number of speakers was included in the Indian English training data (16 speakers for In vs. for example 152 speakers for SAE and 69 for Br). This was probably not enough to build speaker-*independent* HMMs and strong mismatch may have occurred between acoustic models and speakers enrolled for adaptation. We guess that positive effects could also be observed with Indian English if more data and speakers were available for acoustic model training.

7.6 Robustness of SSA under constraining situations

In the previous section, we experimentally verified that a more SV-inclusive phone inventory helps to reduce the amount of shared pronunciations and selected baseforms and generally yields additional performance improvement. In this section, we will analyze the robustness of SSA under two constraining situations, first when the number of adaptation sentences is limited and secondly when speakers do not belong to any of the speech varieties modeled by the system.

7.6.1 Small adaptation data

All of the SSA results presented thus far have been based upon utilizing the full adaptation data set for each test speaker (approx. 153 short utterances on average). Recall that in SSA the adaptation data is used to calculate an estimate of the SVP (Speech Variety Profile) for that speaker. The SVP characterizes the blend of the existing pronunciation models which will be used to form speaker specific pronunciation expectations (*i.e.*, the speaker adapted lexicon). In these experiments, we examined the effect of reducing the available adaptation data on SVP estimation by using only five sentences for adaptation instead of 153. Figure 7.4 gives the average probabilities of each modeled SV using decision trees as pronunciation model and the 164 phone inventory. We see that, on average, using only five short adaptation utterances yields estimated SVPs which are similar to those estimated with the full (153 utterance) adaptation sets. Tests across the different SVs and phone inventories led to similar results.

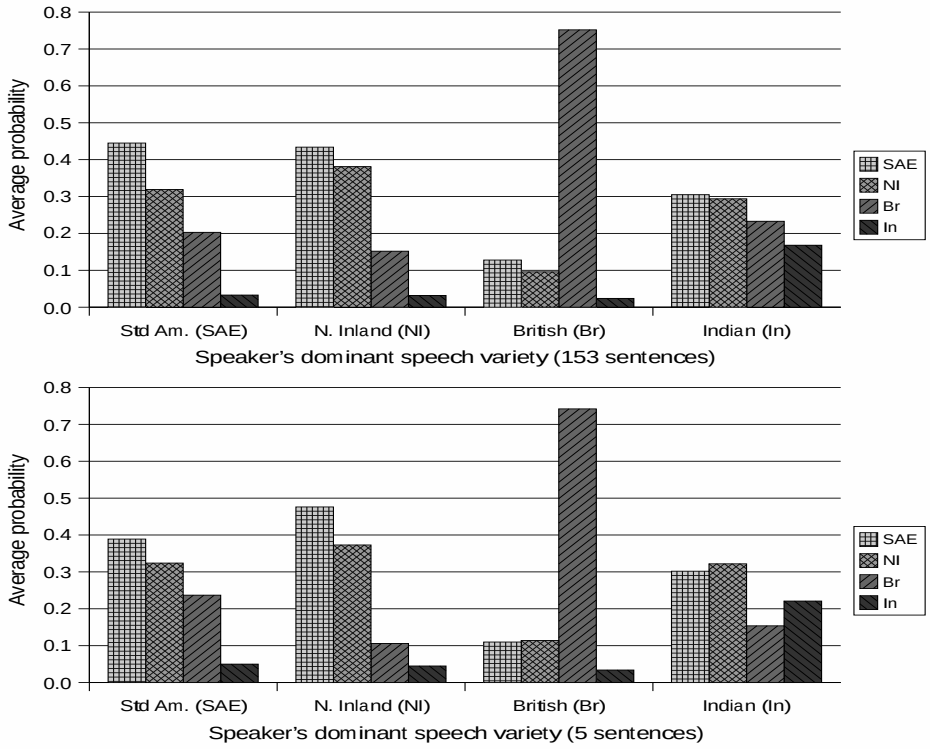

Figure 7.4: Average SVP probabilities for each modeled SV using 153 vs. 5 adaptation sentences

If we are modeling the SV phone inventories well, then we would expect a strong positive correlation between the accuracy of SVP identification and the accuracy of SSA-adapted

ASR. Table 7.12 presents the average WERs obtained for the SVPs derived from the full set of (153) adaptation sentences and from only five sentences. We have found that the SSA method converges to a reasonable characterization of speakers of modeled SVs with very little available data (with even some improvement for the In SV, probably because by chance SSA targeted the In SV slightly better with 5 sentences than with 153 sentences, see Fig. 7.4). Thus, it is suitable for tasks which require rapid adaptation. However, since SVP convergence is solely based on the set of actually occurring phone transformations, results will naturally be more reliable if larger quantities of adaptation data and/or phonetically balanced data are available.

	SAE	NI	Br	In
153 sents	17.70	19.00	23.13	26.73
5 sents	17.82	19.23	23.20	25.57

Table 7.12: SSA results (WER) with 153 vs. 5 adaptation sentences

7.6.2 Non-modeled speech varieties

In section 6.3.1, we claimed that one of the benefits of SSA over a classical SV classification scheme is its flexibility to combine multiple speech varieties to model pronunciation variations of a single speaker. One might rightfully wonder why a blending scheme would be more efficient than a classification scheme. SSA is probably not more efficient than a simple classifier indeed if each speaker's pronunciation style is biased towards a single speech variety and all targeted SVs are correctly modeled. For the sake of verifying it, let us review some of the results of previous sections. We recall that the cheating experiments we described in sections 7.1.4 and 7.5 consisted in completely biasing each speaker's SVP towards his/her dominant speech variety (assuming that it is known in advance); this situation corresponds to the case of an ideal classifier. Table 7.13 compares the SSA and classifier results, first with 41 symbols (results found in Tables 7.2 and 7.3) and then with 164 symbols (results found in Tables 7.10 and 7.11)[6]. We notice that the classifier performs better than SSA, although generally not by much (SSA even shows slightly lower WER with the Br SV). Besides, these results are with an *ideal* classifier; in practice, classifiers are not without errors and hence some performance degradation is expected. Classification and combination schemes are therefore comparable.

	SAE	NI	Br	In	As
Class. - 41 symb.	17.35	20.53	34.68	23.66	28.74
SSA - 41 symb.	17.99	20.86	34.36	23.85	29.36
Class. - 164 symb.	17.22	18.59	23.39	24.40	
SSA - 164 symb.	17.70	19.00	23.13	26.73	

Table 7.13: Comparative results (WER) for handling modeled SVs between ideal classification and SSA's SV blending methods

This section proposes to experimentally show the usefulness of the blending concept, but in a situation where speech varieties of speakers are *not* modeled by the system. For this purpose, the two schemes were again compared: the *classification* scheme that only selects the best SV in adaptation derived SVPs, and the SSA *SV blending* scheme that keeps all SVs

[6]Reminder: no results are available for the As SV with 164 symbols because there was not enough data to train specific As acoustic models

with their respective probabilities. To have a fair comparison, the number of pronunciation variants used for each speaker was made to be approximately the same for both schemes. Fourteen speakers of a diverse group of non-modeled SVs were evaluated. There were two regional / dialectal varieties: African-American, one speaker, and two American speakers with Southern accents. Additionally, three varieties of foreign accented English were represented: Asian, seven speakers, German, two speakers and Spanish, two speakers. Results using the full phone inventory set are given in Table 7.14 and are structured as follows:

Class. shows the results obtained with a classification scheme along with the selected SV in parentheses.

Ideal shows the results obtained with an *ideal* classifier (or an oracle) that always selects the SV that leads to the lowest WER, along with the corresponding SV in parentheses.

SSA shows the results obtained with the SV blending scheme.

	Class.		Ideal		SSA
African-Amer.	13.17	(NI)	12.18	(SAE)	**12.82**
Asian 1	9.78	(NI)	9.06	(SAE)	10.51
Asian 2	26.76	(NI)	26.76	(NI)	27.73
Asian 3	47.69	(Br)	30.00	(NI)	**31.89**
Asian 4	51.42	(NI)	51.42	(NI)	**47.17 ***
Asian 5	34.67	(SAE)	34.67	(SAE)	**33.33 ***
Asian 6	32.14	(Br)	32.14	(Br)	**30.71 ***
Asian 7	34.63	(SAE)	34.63	(SAE)	**33.98 ***
German 1	51.52	(Br)	51.52	(Br)	**51.52 ***
German 2	40.96	(Br)	38.55	(SAE)	**40.96**
South-Amer. 1	35.98	(NI)	35.98	(NI)	**35.15 ***
South-Amer. 2	5.45	(SAE)	3.96	(NI)	**5.45**
Spanish 1	34.15	(SAE)	34.15	(SAE)	**29.27 ***
Spanish 2	38.33	(Br)	26.67	(SAE)	**33.33**
Average	32.62		30.12		**30.27**

Table 7.14: Comparative results (WER) for handling non-modeled SVs between SVP-based classification, the ideal classifier, and SSA's SV blending methods

In the last column, all WERs equal to or lower than the matching classifier scheme counterparts are marked in bold, and those among them that are equal to or lower than the "ideal" classifier WERs are further marked with a '*'. We observe that SSA performs on average better than a classification scheme method (7.2% relative improvement) and is comparable to the "ideal" classifier. Similar behavior can be observed with lower phone inventory HMMs. Therefore and according to the results, SSA is more appropriate to generalize its method to unseen SVs than simple SV classifiers.

7.7 Summary

In this chapter, we experimentally studied the effects of Symbolic Speaker Adaptation (SSA) on ASR using a database with speakers of different speech varieties. The following results were obtained:

- Basic experiments with SSA did not lead to any substantial improvement.
- Although Acoustic Speaker Adaptation (ASA) performed much better than SSA, SSA also yielded bigger improvement when combined with ASA, suggesting that their effects are complimentary.
- Some Viterbi alignment results showed a high proportion of selected pronunciations shared by more than one Speech Variety (SV), which rendered the task of targeting the correct SV(s) more difficult. Especially, a high proportion of baseforms were among the selected pronunciations even though pronunciation styles of Standard American English (SAE) speakers and non-SAE speakers were clearly different. Consequently, addition of new phonetic transcriptions rather added lexical confusability than modeled pronunciation variations.
- More accurate acoustic models (triphones, SV-balanced models) increased the percentage of preference for baseforms during Viterbi alignments, suggesting that true pronunciations were close to baseform transcriptions. However, the ASR system could not target the correct transcriptions with the SAE phone inventory due to new sounds introduced by non-SAE SVs.
- Experiments with more SV-inclusive phone inventory helped to significantly decrease the amount of shared pronunciations and baseforms. Additional improvements with SSA could be observed, especially for the British English with 9.9% relative decrease in WER over the multi-SV trained HMMs (36.9% overall improvement with SV-balanced training). Lack of improvement observed for Indian English was possibly due to a lack of appropriate training data.
- SSA led to similar results with small adaptation data (5 instead of 153 sentences), suggesting that it is suitable for tasks which require rapid adaptation.
- The concept of blending multiple speech varieties was more efficient than classification scheme with speech varieties not modeled by the ASR system. SSA seems therefore more appropriate to generalize the method to speakers of any SV of the same language.

Chapter 8

Conclusion

8.1 Global summary

In this dissertation, we studied the effects of dynamically modeling pronunciation variation. Two aspects were investigated. First, the dynamic approach was applied at two different levels of modeling, lexicon and HMMs. For both levels, generation and/or selection of pronunciation variants were based on the extraction of phonetic features from the input speech. Then, two levels of dynamism were also considered: while the methods based on phonetic features modified the lexicon or acoustic models on a per utterance basis, another method based on symbolic speaker adaptation modified the lexicon on a per speaker basis. A brief description of the different methods and results obtained are described below.

8.1.1 Dynamic lexicon using phonetic features

Chapter 4 proposed a method to generate a lexicon whose pronunciation variants were adapted to each input speech utterance. The technique was based on phonetic features that were extracted automatically from the input speech using a neural network. These features helped to build through several steps a pronunciation network per utterance. These networks were used in two ways: first, they helped to generate alternative pronunciations per word through a two-pass Viterbi alignment in order to build a lexicon augmented with new pronunciation entries; second, they helped to select only the pronunciation variants that were likely present in test utterances by searching them in the corresponding networks, in order to build a specific lexicon per utterance.

All experiments were carried out on the TIMIT database. Each phonetic feature considered separately could be reliably detected from speech and when they were combined, frames were phonetically well identified about half of the time. Further analysis showed that even though phone-based ANNs led to a higher accuracy than feature-based ANNs, the latter generated a smaller confusion distance between the best outputs and their alternatives, suggesting that phonetic features are more suitable as a starting point to generate alternative transcriptions. Pronunciation networks did not however lead to better phone recognition accuracy than canonical transcriptions, but did when combined with the latter. Similarly at the word-level, dynamic lexicons generated from networks did not improve the WER when used alone, but did significantly improve performance once combined with canonical transcriptions. Performance of dynamic lexicons could further be improved by a better generalization of pronunciation

variation to unseen data and a more accurate search algorithm of variants in pronunciation networks, especially with short transcriptions.

8.1.2 Dynamic sharings of Gaussian densities using phonetic features

Chapter 5 extended the dynamic approach to the HMM level. The state-level pronunciation modeling (SLPM) technique introduced by Saraçlar et al. [124] was used for this purpose: when a phoneme could be realized as a distinct phone, the acoustic model of the phoneme shared the Gaussian densities of the phone model to create a hybrid HMM. In chapter 5, this concept was extended to the dynamic case: some phonetic features were extracted from each input utterance during recognition and helped to decide whether a phoneme was likely realized as an alternative phone or not. Gaussian densities were shared only when a phoneme substitution was probable. Deletions and insertions were also modeled using decision trees.

All experiments were carried out again on the TIMIT database. The WER obtained with the dynamic SLPM was comparable to the best result obtained with the traditional SLPM by trying different amounts of Gaussian density sharings. However, the improvement obtained over the baseline system was not statistically significant in any of the two cases. Combination of decision trees and phonetic features helped to significantly decrease the PER, but surprisingly increased the WER. An independent section was dedicated to the detection of phonetic features in spontaneous speech. Two different feature systems (SPE and multi-valued) were evaluated and led to similar overall results. Only small degradation per single feature was generally observed compared to the results obtained with read speech, but which led to a much higher frame-level degradation (from above 50% to below 40% correct rate) once the features were combined.

8.1.3 Symbolic speaker adaptation

While the previous methods dynamically modified the lexicon or acoustic models on a per utterance basis, chapter 6 presented a method called symbolic speaker adaptation (SSA) to implement a pseudo-dynamic approach on a per speaker basis. The general idea was that any speaker's pronunciation could be represented by a combination of several pronunciation styles (called "speech varieties", SV) modeled by the ASR system. The objective of the adaptation was to create a speech variety profile for each speaker with the relative importance of each SV that best reflected his/her pronunciation characteristics. These profiles influenced how the canonical lexicon was expanded with pronunciation variants, so that each speaker's pronunciation was modeled by a different lexicon.

The concept was experimented in chapter 7 with multiple dialects and foreign accents as speech varieties. Basic experiments with SSA did not lead to any significant improvement. However, SSA did perform better when combined with standard acoustic speaker adaptation (ASA), suggesting that SSA and ASA are complimentary. Analysis of intermediate results revealed that the majority of preferred pronunciations during the adaptation process were baseforms, even though there were clearly pronunciation variations in the evaluated SVs. Expansion of the initial phone inventory with more SV-inclusive phones helped to significantly decrease this preference for baseforms and helped to gain additional improvement in most cases. Furthermore, it was experimentally shown that SSA was able to hone in on a speaker's speech variety with small amounts of adaptation data and that the SV-blending scheme of SSA better modeled unknown speech varieties than a standard SV-classification scheme.

8.2 Contributions of this dissertation

Dynamic pronunciation modeling at phonetic level [100]: some contributions already exist in this area, but consist of modifying pronunciation probabilities through N-best list or lattice rescoring. This dissertation studied the effects of completely accepting or rejecting pronunciation variants in the ASR lexicon.

Dynamic pronunciation modeling at acoustic level [101]: based on state-level pronunciation modeling, a method was proposed to modify HMMs so that they dynamically account for partial pronunciation changes.

Incorporation of articulatory knowledge into pronunciation modeling [100][101]: a method was proposed to create pronunciation networks from a set of phonetic (articulatory) features extracted from the input speech utterance. Phonetic features served also as cues to dynamically select the most likely phonetic transcriptions or create HMMs during recognition.

Detection of phonetic features in read vs. spontaneous speech : several papers published accuracy of phonetic feature detection in read speech, but applicability of their method to spontaneous speech was not clearly reported. This dissertation applied a same method to both read and spontaneous speech to directly measure the degradation implied. Furthermore, two phonetic feature systems (SPE and multi-valued) were evaluated and compared to a more traditional phone-based system.

Symbolic speaker adaptation [98][99]: a new method to model pronunciation variation at the speaker level was proposed. In particular, this technique is well-suited for modeling multiple speech varieties (dialects and foreign accents) simultaneously.

8.3 Some directions for future work

Concerning the dynamic lexicon approach using phonetic features, some suggestions to ameliorate some specific parts of the system have already been listed in section 4.9. Apart from these system-specific points, an aspect to be determined is how a dynamic approach would perform in spontaneous speech. Since the latter includes a lot more variations than read speech, a dynamic method could bring bigger improvement due to its better pronunciation coverage than a standard (static) method. But first, the cues used to dynamically select the set of pronunciation variants need to be estimated reliably, which was not the case of the phonetic features once they were combined (cf. section 5.7). Furthermore, this dissertation was centered on articulatory positions to predict the possible pronunciations, but additional cues (*e.g.*, speaking rate, prosody), could be helpful to get more reliable estimates.

The lack of WER improvement with dynamic acoustic models despite the better phone accuracy achieved is rather puzzling. Recently, Yi and Fung [157] evaluated the state-level pronunciation modeling (SLPM) technique on the Mandarin Broadcast News corpus and also found that it was of little benefit. They proposed instead to align baseform and surface form transcriptions and to create an additional partial change phone model (PCPM) each time a phoneme was frequently mapped to a distinct phone. Then, to increase the robustness, they created a "conventional" tree for the canonical model and an "auxiliary" tree for each related PCPM to state-tie the different possible triphone contexts, and finally merged the leaves of the conventional and auxiliary trees based on minimum Gaussian distance to allow canonical

models to capture partial pronunciation changes. They obtained an additional 1.45% absolute syllable error rate reduction compared to SLPM. It would be interesting to see if a dynamic approach based on this method could further improve performance.

A possible extension of the symbolic speaker adaptation method would be to explore the selection of an optimal speech variety (SV) set to model. For example, in chapter 7, we had modeled Standard American English and Northern Inland (*e.g.*, English spoken in Chicago) as separate SVs; this is obviously not an appropriate choice since their acoustic and phonetic properties are very similar. We believe that a data-driven approach to determine an optimal set of "canonical bases" that are not necessarily bound to any particular SV could be advantageous, for example using a concept similar to Kuhn et al. [93], but at the SV level. It is interesting to note that Goronzy [55] (chapter 9) also recently proposed a similar idea as future work and visualized a set of rules in an eigenpronunciation space. Furthermore, the adaptation method presented in this dissertation worked under an offline mode, that is, once a lexicon was created for a speaker after a separate adaptation process, its content did not change any more later on. Ideally, adaptation should be done in an online (incremental and unsupervised) fashion: if the system detects that a speaker's pronunciation has changed from a previous session, pronunciation variants in the lexicon should be updated accordingly. However, an unsupervised mode implies a risk of using erroneous transcriptions for adaptation that could worsen the ASR performance. Care must therefore be taken so as to insure a good robustness in this regard.

Appendix A

Phone inventory

The following table shows the original phone set of the TIMIT database. Phones also belonging to the Myosphere inventory have been marked in bold; they correspond to a reduced set of 39 phones. Please note that these symbols were slightly modified for the experiments in order to be compatible with the phone-features conversion tables used. The relevant modifications are described in appendix B.

Phone	Example	Phone	Example	Phone	Example
aa	Bob	eng	camping	**oy**	boy
ae	bat	**er**	bird	**p**	potholder
ah	but	**ey**	bait	pcl	stop ([p] closure)
ao	bought	**f**	fat	q	(glottal stop)
aw	down	**g**	game	**r**	rent
ax	about	gcl	game ([g] closure)	**s**	sat
ax-h	(voiceless [ax])	**hh**	head	**sh**	shut
axr	butter	hv	(voiced [hh])	**t**	ten
ay	buy	**ih**	bit	tcl	streetcar ([t] closure)
b	boat	ix	animal (centralized [ih])	**th**	thing
bcl	bobtail ([b] closure)	**iy**	beat	**uh**	book
ch	church	**jh**	judge	**uw**	boot
d	dock	**k**	cot	ux	suit
dcl	bloodclot ([d] closure)	kcl	clockwork ([k] closure)	**v**	vat
dh	that	**l**	let	**w**	wit
dx	(flap or tap [t])	**m**	met	**y**	you
eh	bet	**n**	net	**z**	zoo
el	battle	**ng**	sing	**zh**	azure
em	bottom	nx	(nasal flap)	**pau**,epi	(pause)
en	button	**ow**	show	**sil**,#h,h#	(silence)

Appendix B

Phone-features conversion tables

B.1 SPE feature system 1

The table in the next page lists the phones and corresponding SPE features given by King and Taylor [82] (phones with same feature combinations have been merged in this version) and used in chapter 4. The alternative '0' values in the table correspond to the ternary version used in section 4.8.1; these features were considered as not relevant for the corresponding phones (partly based on the ternary feature table given by Brondsted [17], cf. appendix B.2). Please note that some non-relevant features according to phonological theories have not been marked in order to limit the implied reduction in discrimination between consonants and between vowels compared to the original binary version. The following abbreviations were used for the phonetic features:

voc:	vocalic	**cns**:	consonantal
hgh:	high	**bck**:	back
low:	low	**ant**:	anterior
cor:	coronal	**rnd**:	round
tns:	tense	**voi**:	voice
cnt:	continuant	**nas**:	nasal
str:	strident	**sil**:	silence

Phone	voc	cns	hgh	bck	low	ant	cor	rnd	tns	voi	cnt	nas	str	sil
aa	+	-	-	+	+	-/0	-/0	-	+	+/0	+/0	-/0	-/0	-
ae	+	-	-	-	+	-/0	-/0	-	+	+/0	+/0	-/0	-/0	-
ah_ax_axh	+	-	-	+	-	-/0	-/0	-	-	+/0	+/0	-/0	-/0	-
ao_aw	+	-	-	+	+	-/0	-/0	+	+	+/0	+/0	-/0	-/0	-
axr_er_ih	+	-	-	-	-	-/0	-/0	-	-	+/0	+/0	-/0	-/0	-
ay_oy	-	-	-	+	+	-/0	-/0	-	+	+/0	+/0	-/0	-/0	-
b	-	+	-	-/0	-/0	+	-	-/0	-/0	+	-	-	-	-
bcl_pcl	-	+	-	-/0	-/0	+	-	-/0	-/0	-	-	-	-	-
ch	-	+	-	+/0	-/0	-	+	-/0	-/0	-	-	-	+	-
d	-	+	+	-/0	-/0	+	+	-/0	-/0	+	-	-	-	-
dcl_t_tcl	-	+	-	-/0	-/0	+	+	-/0	-/0	-	-	-	-	-
dh_el_th	-	+	-	-/0	-/0	+	+	-/0	-/0	+	+	-	-	-
dx	+	+	-	-/0	-/0	+	+	-/0	-/0	+	-	-	-	-
eh	-	-	-	-	-	-/0	-/0	-	-	+/0	+/0	-/0	-/0	-
em_m_n	-	+	-	-/0	-/0	+	-	-/0	-/0	+	-	+	-	-
en	-	+	-	-/0	-/0	+	+	-/0	-/0	+	-	+	-	-
eng_ng	-	+	-	+/0	-/0	-	-	-/0	-/0	+	-	+	-	-
ey	-	-	-	-	-	-/0	-/0	-	+	+/0	+/0	-/0	-/0	-
f	-	+	-	-/0	-/0	+	-	-/0	-/0	-	+	-	+	-
g	-	+	-	+/0	-/0	-	-	-/0	-/0	+	+	-	-	-
gcl	-	+	-	+/0	-/0	-	-	-/0	-/0	-	-	-	-	-
hh	-	+	-	-	+	-	-	-/0	-/0	-	+	-	-	-
hv	+	+	-	-	+	-	-	-/0	-/0	+	+	-	-	-
ix	+	-	+	-	-	-/0	-/0	-	-	+/0	+/0	-/0	-/0	-
iy	-	-	+	-	-	-/0	-/0	-	+	+/0	+/0	-/0	-/0	-
jh	-	+	+	-/0	-/0	-	+	-/0	-/0	+	-	-	+	-
k_kcl	-	+	+	+/0	-/0	-	-	-/0	-/0	-	-	-	-	-
l	-	+	+	-	-/0	+	+	-/0	-/0	+	+	-	-	-
nx	+	+	+	-/0	-/0	+	+	-/0	-/0	+	-	+	-	-
ow	+	-	-	+	-	-/0	-/0	+	+	+/0	+/0	-/0	-/0	-
p	-	+	+	-/0	-/0	+	-	-/0	-/0	-	-	-	-	-
q	-	+	-	+/0	+/0	-	-	-/0	-/0	-	-	-	-	-
r	+	+	-	-	-/0	-	+	-/0	-/0	+	+	-	-	-
s	-	+	-	-/0	-/0	+	+	-/0	-/0	-	+	-	+	-
sh	-	+	+	-/0	-/0	-	+	-/0	-/0	-	+	-	+	-
uh	+	-	+	+	-	-/0	-/0	-	-	+/0	+/0	-/0	-/0	-
uw_ux	+	-	+	+	-	-/0	-/0	+	+	+/0	+/0	-/0	-/0	-
v	-	+	-	-/0	-/0	+	-	-/0	-/0	+	+	-	+	-
w	-	+	+	-	-	-	-	+	-	+/0	+/0	-/0	-/0	-
y	-	+	+	-	-	-	-	-	-	+/0	+/0	-/0	-/0	-
z	-	+	-	-/0	-/0	+	+	-/0	-/0	+	+	-	+	-
zh	-	+	+	-/0	-/0	-	+	-/0	-/0	+	+	-	+	-
sil	-	-	-	-	-	-	-	-	-	-	-	-	-	+

B.2 SPE feature system 2

The table in the next page lists the phones and corresponding SPE features given by Brondsted [17] and used in chapter 5. The '(+)' and '(-)' values in the table correspond to the binary version used in section 5.5.3 to train the ANN (partly based on the binary feature table given by King and Taylor [82], cf. appendix B.1); they were considered as non-relevant features in the original version. The following abbreviations were used for the phonetic features:

snr:	sonorant	**syl**:	syllabic
cns:	consonantal	**hgh**:	high
bck:	back	**low**:	low
fnt:	front	**ant**:	anterior
cor:	coronal	**rnd**:	round
voi:	voice	**cnt**:	continuant
nas:	nasal	**str**:	strident
sil:	silence		

Some modifications were brought to the original TIMIT phone inventory in compliance with the replacements proposed by Brondsted and are listed below:

Polyphonematic replacements:

aw	→	ah w
ay	→	ah y
ey	→	eh y
iy	→	ih y
ow	→	oh w
oy	→	ao y
uw	→	uh w

Monophonematic replacements:

bcl b	→	b
dcl d	→	d
gcl g	→	g
kcl k	→	k
pcl p	→	p
tcl t	→	t

Allophone-phoneme replacements:

ax-h	→	ax
dx	→	d
hv	→	hh
nx	→	n
q	→	t
ux	→	uw

Phone	snr	syl	cns	hgh	bck	low	fnt	ant	cor	rnd	voi	cnt	nas	str	sil
aa	+	+	-	-	+	-	+	-	(-)	(-)	(+)	(+)	(-)	(-)	-
ae	+	+	-	-	-	+	+	-	(-)	(-)	(+)	(+)	(-)	(-)	-
ah	+	+	-	-	+	-	-	-	(-)	(-)	(+)	(+)	(-)	(-)	-
ao	+	+	-	-	+	-	+	+	(-)	(-)	(+)	(+)	(-)	(-)	-
ax	+	+	-	-	-	-	-	-	(-)	(-)	(+)	(+)	(-)	(-)	-
axr	+	+	-	-	-	-	+	-	(-)	(-)	(+)	(+)	(-)	(-)	-
b	-	-	+	-	(-)	(+)	(-)	(-)	+	-	+	-	-	-	-
ch	-	-	+	+	(+)	(-)	(-)	(-)	-	+	-	-	-	+	-
d	-	-	+	-	(-)	(-)	(-)	(-)	+	+	+	-	-	-	-
dh	-	-	+	-	(-)	(-)	(-)	(-)	+	+	+	+	-	-	-
eh	+	+	-	-	-	+	-	-	(-)	(-)	(+)	(+)	(-)	(-)	-
el	+	+	+	-	-	-	-	(-)	+	+	+	+	-	-	-
em	+	+	+	-	(-)	(-)	-	(-)	+	-	+	-	+	-	-
en	+	+	+	-	(-)	(-)	-	(-)	+	+	+	-	+	-	-
eng	+	+	+	+	(+)	(-)	-	(-)	-	-	+	-	+	-	-
er	+	+	-	-	-	-	-	+	(-)	(-)	(+)	(+)	(-)	(-)	-
f	-	-	+	-	(-)	(+)	(-)	(-)	+	-	-	+	-	+	-
g	-	-	+	+	(+)	(-)	(-)	(-)	-	-	+	-	-	-	-
hh	+	-	-	-	-	-	+	(-)	-	-	-	+	-	-	-
ih	+	+	-	+	-	+	-	-	(-)	(-)	(+)	(+)	(-)	(-)	-
ix	+	+	-	+	-	-	-	-	(-)	(-)	(+)	(+)	(-)	(-)	-
jh	-	-	+	+	(-)	(-)	(-)	(-)	-	+	+	-	-	+	-
k	-	-	+	+	(+)	(-)	(-)	(-)	-	-	-	-	-	-	-
l	+	-	+	-	-	-	-	(-)	+	+	+	+	-	-	-
m	+	-	+	-	(-)	(+)	-	(-)	+	-	+	-	+	-	-
n	+	-	+	-	(-)	(-)	-	(-)	+	+	+	-	+	-	-
ng	+	-	+	+	(+)	(-)	-	(-)	-	-	+	-	+	-	-
oh	+	+	-	-	+	-	-	+	(-)	(-)	(+)	(+)	(-)	(-)	-
p	-	-	+	-	(-)	(+)	(-)	(-)	+	-	-	-	-	-	-
r	+	-	+	-	-	-	-	(-)	-	+	+	+	-	-	-
s	-	-	+	-	(-)	(-)	(-)	(-)	+	+	-	+	-	+	-
sh	-	-	+	+	(-)	(-)	(-)	(-)	-	+	-	+	-	+	-
t	-	-	+	-	(-)	(-)	(-)	(-)	+	+	-	-	-	-	-
th	-	-	+	-	(-)	(-)	(-)	(-)	+	+	-	+	-	-	-
uh	+	+	-	+	+	-	-	+	(-)	(-)	(+)	(+)	(-)	(-)	-
v	-	-	+	-	(-)	(+)	(-)	(-)	+	-	+	+	-	+	-
w	+	-	-	+	+	-	-	+	-	-	(+)	(+)	(-)	(-)	-
y	+	-	-	+	-	+	-	-	-	-	(+)	(+)	(-)	(-)	-
z	-	-	+	-	(-)	(-)	(-)	(-)	+	+	+	+	-	+	-
zh	-	-	+	+	(-)	(-)	(-)	(-)	-	+	+	+	-	+	-
sil	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+

B.3 Multi-valued feature system

The table in the next page lists the phones and corresponding multi-valued features partly based on Kirshenbaum [87] and used in section 5.7. The following abbreviations were used for the phonetic features:

Voicing:

vcd:	voiced	**vls**:	voiceless

Place:

blb:	bilabial	**lbd**:	labio-dental
dnt:	dental	**alv**:	alveolar
pla:	palato-alveolar	**pal**:	palatal
vel:	velar	**lbv**:	labio-velar
glt:	glottal		

Manner:

stp:	stop	**frc**:	fricative
nas:	nasal	**apr**:	approximant
lat:	lateral		

Height:

hgh:	high	**smh**:	semi-high
umd:	upper-mid	**lmd**:	lower-mid
low:	low		

Front-back:

fnt:	front	**cnt**:	center
bck:	back		

Rounding:

unr:	unrounded	**rnd**:	rounded

Special:

nil:	non-relevant	**sil**:	silence

The following polyphonematic replacements were also applied to the original Myosphere phone inventory:

aw	→	a uh
ay	→	a ih
ey	→	eh iy
ow	→	oh uh
oy	→	ao iy
ch	→	t sh
jh	→	d zh
er	→	ah r

Phone	Voicing	Place	Manner	Height	Fnt-bck	Rnding
a	nil	nil	nil	low	cnt	unr
aa	nil	nil	nil	low	bck	unr
ae	nil	nil	nil	low	fnt	unr
ah	nil	nil	nil	lmd	cnt	unr
ao	nil	nil	nil	lmd	bck	rnd
b	vcd	blb	stp	nil	nil	nil
d	vcd	alv	stp	nil	nil	nil
dh	vcd	dnt	frc	nil	nil	nil
eh	nil	nil	nil	lmd	fnt	unr
f	vls	lbd	frc	nil	nil	nil
g	vcd	vel	stp	nil	nil	nil
hh	vls	glt	apr	nil	nil	nil
ih	nil	nil	nil	smh	fnt	unr
iy	nil	nil	nil	hgh	fnt	unr
k	vls	vel	stp	nil	nil	nil
l	vcd	alv	lat	nil	nil	nil
m	vcd	blb	nas	nil	nil	nil
n	vcd	alv	nas	nil	nil	nil
ng	vcd	vel	nas	nil	nil	nil
oh	nil	nil	nil	umd	bck	rnd
p	vls	blb	stp	nil	nil	nil
r	vcd	alv	apr	nil	nil	nil
s	vls	alv	frc	nil	nil	nil
sh	vls	pla	frc	nil	nil	nil
t	vls	alv	stp	nil	nil	nil
th	vls	dnt	frc	nil	nil	nil
uh	nil	nil	nil	smh	bck	rnd
uw	nil	nil	nil	hgh	bck	rnd
v	vcd	lbd	frc	nil	nil	nil
w	vcd	lbv	apr	nil	nil	nil
y	vcd	pal	apr	nil	nil	nil
z	vcd	alv	frc	nil	nil	nil
zh	vcd	pla	frc	nil	nil	nil
sil	sil	sil	sil	sil	sil	sil

Appendix C

HMM training procedures

The following sections detail the procedures applied to build HMM-based ASR systems used as baselines with the TIMIT and Myosphere databases.

C.1 HMM training procedure for TIMIT

39 monophone models were trained using the TIMIT database and were used as the baseline system for the experiments described in section 5.5. Each model had three left-to-right emitting states and no skip of a state was allowed. All states were modeled with ten mixtures of Gaussian means and diagonal covariance matrixes. They were trained with acoustic vectors containing 39 Mel-frequency cepstral coefficients (MFCC): 12 static coefficients and 1 normalized log energy, plus their corresponding delta and acceleration coefficients. These acoustic vectors were obtained from the input speech using a Hamming window of 25ms and a 10ms frame interval. The coding parameters used with HTK are provided below (many values are the same as suggested in the tutorial example of the HTK book [158]):

```
SOURCEFORMAT = NIST
TARGETKIND = MFCC_E_D_A
TARGETRATE = 100000.0
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = T
```

Besides the regular phone models, additional models for "silence" (sil) and "short pause" (sp) were also added. The "sil" model had exactly the same topology than a regular phone model. The "sp" model on the other hand had only one emitting state tied with the center state of the "sil" model; the state could be skipped via an alternative short cut link. All models were trained using the manually labeled and segmented transcriptions provided by TIMIT.

The training procedure respected the following steps (the relevant HTK commands are

between parentheses):

Initialization (HInit) : a first uniform segmentation of the training data initialized the HMM parameters of the 39 phones + 1 sil (no "sp" model yet), followed by several iterations of Viterbi alignments and parameter updates.

Isolated unit re-estimation (HRest) : the initialized parameters were re-estimated for each HMM separately using the Baum-Welch algorithm, given the relevant speech segments extracted from the training data for the HMM.

Addition of short pause : the "sp" model was added to the set of HMMs following the description above. Time boundary information at both word and phone levels to compare in order to add an "sp" label after each word in phonetic transcriptions.

Embedded training (HERest), 4 iterations : parameters of all HMMs were re-estimated simultaneously by applying the Baum-Welch algorithm over each training utterance using the corresponding sequence of phone models.

Mixture splitting (HHEd), 1 $\rightarrow$ 10 : the number of Gaussian mixtures was progressively increased, with 4 embedded Baum-Welch re-estimations between two consecutive splits.

Since the number of phones was reduced compared to the original TIMIT inventory, the phone labels used for training were modified accordingly (the list of replacements can be found in appendix B.2). Concerning the associated time intervals (required for the isolated unit re-estimation, HRest), when two monophthongs were replaced by one diphthong, the corresponding time intervals were concatenated; when a diphthong was replaced by two monophthongs, the initial time interval was divided by two in the middle.

C.2 HMM training procedure for Myosphere

Like with TIMIT, 39 monophone models were trained using the Myosphere database and were used as the baseline system for the experiments described in chapter 7. Each model had three left-to-right emitting states and no skip of a state was allowed. All states were modeled with five mixtures of Gaussian means and diagonal covariance matrixes. They were trained with acoustic vectors containing 39 Mel-frequency cepstral coefficients (MFCC): 12 static coefficients and 1 normalized log energy, plus their corresponding delta and acceleration coefficients. These acoustic vectors were obtained from the input speech using a Hamming window of 25ms and a 10ms frame interval. In addition to that, a cepstral mean normalization [5] was applied to the coefficients to remove channel distortion effects. The coding parameters used with HTK are provided below (many values are the same as suggested in the tutorial example of the HTK book [158]):

```
SOURCEFORMAT = NIST
TARGETKIND = MFCC_E_D_A_Z
TARGETRATE = 100000.0
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 20
```

```
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = T
```

Besides the regular phone models, additional models for "silence" (sil) and "short pause" (sp) were also added. The "sil" model had similar topology than a regular phone model (3 left-to-right HMMs), except that a bidirectional link between the first and the last emitting state was added, with the objective of better absorbing impulsive noises in the training data. The "sp" model had only one emitting state tied with the center state of the "sil" model; the state could be skipped via an alternative short cut link.

In contrast with TIMIT, no manually labeled transcription was available in the Myosphere database (a small subset was actually transcribed, but was too small). As an alternative, the canonical transcriptions of reference words were used initially, but they were followed by Viterbi alignments every 5 Baum-Welch iterations using a lexicon with multiple pronunciations to re-estimate the labels during the training.

The training procedure respected the following steps (the relevant HTK commands are between parentheses):

Flat start (HCompV) : a global mean and a global covariance were calculated from the training data and were set as initial means and covariances of all models ("sp" model not included yet).

Embedded training (HERest), 5 iterations : parameters of all HMMs were re-estimated simultaneously by applying the Baum-Welch algorithm over each training utterance using the canonical transcriptions of reference words (the transcriptions were taken from the "BLex1" lexicon mentioned in section 7.1.2).

Addition of short pause : the "sp" model was added to the set of HMMs following the description above. The lexicon was modified accordingly: each phonetic transcription in the lexicon was expanded into two versions, one with the "sp" label (reminder: the "sp" model could be skipped) and another with the "sil" label to account for longer inter-word silences.

Label re-estimation (HVite) + embedded training (HERest), 5 iterations : the initial transcriptions were re-estimated using Viterbi alignment and a lexicon with multiple pronunciations (BLex2, cf. section 7.1.2). Then, the acoustic models were re-estimated using the new labels. These two processes were re-iterated until convergence of word accuracy on a cross-validation data.

Mixture splitting (HHEd), 1 $\rightarrow$ 5 : the number of Gaussian mixtures was progressively increased, with 4 embedded Baum-Welch re-estimations between two consecutive splits.

Appendix D

Splitting questions for decision trees

The following table shows the list of questions used to build the decision trees in section 5.6.2. All questions await a "yes" or "no" answer and their associated phones correspond to the "yes" answer. Questions were based on the example provided by HTK (version 2.2, file "RMHTK_V2.2/lib/quests.hed") for tree-based clustering, whereas their associated phones were selected to fit with the phone-features conversion table shown in appendix B.2.

Question	Phones
Silence ?	sil
Consonant ?	b, ch, d, dh, el, em, en, eng, f, g, hh, jh, k, l, m, n, ng, p, r, s, sh, t, th, v, w, y, z, zh
Vowel ?	aa, ae, ah, ao, ax, axr, eh, er, ih, ix, oh, uh
Stop ?	b, d, g, k, p, t
Nasal ?	em, en, eng, m, n, ng
Fricative ?	ch, dh, hh, f, jh, s, sh, th, v, z, zh
Affricate ?	ch, jh
Liquid ?	el, l, r, w, y
Front ?	b, f, m, p, v, y, ae, eh, ih
Central ?	d, dh, el, em, en, hh, jh, l, n, r, s, sh, t, th, z, zh, ax, axr, er, ix
Back ?	ch, eng, g, k, ng, w, aa, ah, ao, oh, uh
C-Front ?	b, f, m, p, v, y
C-Central ?	d, dh, el, em, en, hh, jh, l, n, r, s, sh, t, th, z, zh
C-Back ?	ch, eng, g, k, ng, w
V-Front ?	ae, eh, ih
V-Central ?	ax, axr, er, ix
V-Back ?	aa, ah, ao, oh, uh
Fortis ?	ch, f, k, p, s, sh, t, th
Lenis ?	b, d, dh, g, jh, sh, v, z, zh
UnFortLenis ?	el, em, en, eng, hh, l, m, n, ng, r, w, y
Coronal ?	ch, d, dh, el, en, jh, l, n, r, s, sh, t, th, z, zh
NonCoronal ?	b, em, eng, f, g, hh, k, m, ng, p, v, w, y
Anterior ?	b, d, dh, el, em, en, f, l, m, n, p, s, t, th, z, v
NonAnterior ?	ch, eng, g, hh, jh, k, ng, r, sh, w, y, zh

(continued next page)

(continued)	
Continuent ?	dh, el, f, hh, l, r, s, sh, th, v, w, y, z, zh
NonContinuent ?	b, ch, d, em, en, eng, g, jh, k, m, n, ng, p, t
Strident ?	ch, f, jh, s, sh, v, z, zh
NonStrident ?	b, d, dh, el, em, en, eng, g, hh, k, l, m, n, ng, p, r, t, th, w, y
Glide ?	el, hh, l, r, w, y
Syllabic ?	el, em, en, eng
Unvoiced-Cons ?	ch, f, hh, k, p, s, sh, t, th
Voiced-Cons ?	b, d, dh, el, em, en, eng, g, jh, l, m, n, ng, r, v, w, y, z, zh
Unvoiced-All ?	ch, f, hh, k, p, s, sh, t, th, sil
Long ?	aa, ae, ao, er
Short ?	ah, ax, axr, eh, ih, ix, oh, uh
High ?	ih, ix, uh
Medium ?	ah, ax, eh, er, oh
Low ?	aa, ae, ao, axr
Rounded ?	ao, er, oh, uh, w
Unrounded ?	aa, ae, ah, ax, axr, eh, ih, ix, y
AVowel ?	aa, ae, ah, ao, ax, axr, er
EVowel ?	ae, eh
IVowel ?	ih, ix
OVowel ?	ao, oh
UVowel ?	uh
Voiced-Stop ?	b, d, g
Unvoiced-Stop ?	k, p, t
Front-Stop ?	b, p
Central-Stop ?	d, t
Back-Stop ?	g, k
Voiced-Fric ?	dh, jh, v, z, zh
Unvoiced-Fric ?	ch, f, hh, s, sh, th
Front-Fric ?	f, v
Central-Fric ?	dh, hh, jh, s, sh, th, z, zh
Back-Fric ?	ch
aa ?	aa
ae ?	ae
(...)	(...)
z ?	z
zh ?	zh

Appendix E

Statistical significance test

The confidence intervals used in this dissertation for statistical significance tests were already used previously in other works (*e.g.*, [102], [153]). It is assumed that the probability to recognize a unit correctly follows a binomial distribution. Supposing that n is the number of units (*e.g.*, words) in the test set and p is the probability of correctly recognizing an unit (*i.e.*, given by the accuracy estimate of the assessed ASR system), the probability of correctly matching k units is given by:

$$\binom{n}{k} \cdot p^k \cdot (1-p)^{(n-k)} \tag{E.1}$$

Under this assumption, the binomial distribution can be approximated by a Gaussian distribution with mean np and variance $np(1-p)$. A confidence interval with low and high boundaries c_{min} and c_{max} can then be defined so that the probability that the exact accuracy $\tilde{p}$ is inside this interval is $1-\alpha$, where α is a value associated with the significance level:

$$(c_{min} < \tilde{p} < c_{max}) = 1 - \alpha \tag{E.2}$$

The boundaries of the confidence interval are determined by the following expression:

$$(c_{max}, c_{min}) = \frac{2np + z_\alpha \pm z_\alpha \sqrt{4np(1-p) + z_\alpha^2}}{2(n + z_\alpha^2)} \tag{E.3}$$

where $z_\alpha = 1.96$ for $\alpha = 0.05$ (95% confidence). In case error rates are used instead of accuracies, the corresponding boundaries of confidence intervals are obtained by simply taking $1 - c_{min}$ and $1 - c_{max}$.

In this dissertation, a change in performance due to the application of a method to an ASR system was assumed to be statistically significant when the confidence intervals associated with the word error rates before and after application of the method did not overlap.

Bibliography

[1] M. Adda-Decker and L. Lamel. Pronunciation variants across system configuration, language and speaking style. *Speech Communication*, 29(2-4):83–98, 1999.

[2] I. Amdal. *Learning pronunciation variation - a data-driven approach to rule-based lexicon adaptation for automatic speech recognition.* PhD thesis, Norwegian university of science and technology, 2002.

[3] I. Amdal, F. Korkmazskiy, and A.C. Surendran. Joint pronunciation modelling of non-native speakers using data-driven methods. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, 2000.

[4] L.M. Arslan and H.L. Hansen. Frequency characteristics of foreign accented speech. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 97)*, Munich, Germany, 1997.

[5] B. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustical Society of America*, 55(6):1304–1312, 1974.

[6] B.S. Atal, J.J. Chang, M.V. Mathews, and J.W. Turkey. Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer sorting technique. *Journal of the Acoustical Society of America*, 63:1535–1555, 1978.

[7] M. Bacchiani and M. Ostendorf. Joint lexicon, acoustic unit inventory and model design. *Speech Communication*, 29(2-4):99–114, 1999.

[8] G. Bailly, C. Abry, L.-J. Boe, R. Laboissiere, P. Perrier, and J.-L. Schwartz. Inversion and speech recognition. In *Proceedings of the 6th European Signal Processing Conference (EUSIPCO 92)*, Brussels, Belgium, 1992.

[9] J.K. Baker. The DRAGON system approach - an overview. *IEEE transactions on acoustics, speech and signal processing*, 23:24–29, 1975.

[10] V. Beattie, S. Edmondson, D. Miller, Y. Patel, and G. Talvola. An integrated multi-dialect speech recognition system with optional speaker adaptation. In *Proceedings of the 4th European Conference on Speech Communication and Technology (Eurospeech 95)*, Madrid, Spain, 1995.

[11] BEEP Pronunciation Dictionary.
URL: http://svr-www.eng.cam.ac.uk/comp.speech/Section1/Lexical/beep.html.

[12] N.N. Bitar and C.Y. Espy-Wilson. Speech parameterization based on phonetic features: application to speech recognition. In *Proceedings of the 4th European Conference on Speech Communication and Technology (Eurospeech 95)*, Madrid, Spain, 1995.

[13] N.N. Bitar and C.Y. Espy-Wilson. Knowledge-based parameters for HMM speech recognition. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 96)*, Atlanta, USA, 1996.

[14] N.N. Bitar and C.Y. Espy-Wilson. The design of acoustic parameters for speaker-independent speech recognition. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech 97)*, Rhodes, Greece, 1997.

[15] C.S. Blackburn, J.P. Vonwiller, and R.W. King. Automatic accent classification using artificial neural networks. In *Proceedings of the 3rd European Conference on Speech Communication and Technology (Eurospeech 93)*, Berlin, Germany, 1993.

[16] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and regression trees.* Wadsworth, Belmont, 1984.

[17] T. Brondsted. A SPE based distinctive feature composition of the CMU label set in the TIMIT database. Technical Report IR 98-1001, Center for PersonKommunikation, Aalborg University, 1998.

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

[19] J. Carson-Berndsen and M. Walsh. Defining constraints for multilinear speech processing. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg, Denmark, 2001.

[20] R. Carter and D. Nunan. *The Cambridge guide to teaching English to speakers of other languages.* Cambridge University Press, Cambridge, UK, 2001.

[21] D. Caseiro, F.M. Silva, I. Trancoso, and C. Viana. Automatic alignment of MAP task dialogs using WFSTs. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[22] S. Chang, S. Greenberg, and M. Wester. An elitist approach to articulatory-acoustic feature classification. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg, Denmark, 2001.

[23] S. Chang, L. Shastri, and S. Greenberg. Automatic phonetic transcription of spontaneous speech (American English). In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, 2000.

[24] C. Chesta, A. Girardi, and P. Laface. Discriminative training of Hidden Markov Models. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 98)*, Seattle, USA, 1998.

[25] N. Chomsky and M. Halle. *The sound pattern of English.* Harper & Row, New York, USA, 1968.

[26] CMU Pronunciation Dictionary. URL: http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[27] M. Cohen, H. Murveit, J. Bernstein, P. Price, and M. Weintraub. The Decipher speech recognition system. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 90)*, Albuquerque, USA, 1990.

[28] M.H. Cohen. *Phonological structures for speech recognition.* PhD thesis, University of California, Berkeley, 1989.

[29] N. Cremelie and J.-P. Martens. In search of better pronunciation models for speech recognition. *Speech Communication*, 29(2-4):115–136, 1999.

[30] A. Cruttenden. *Gimson's pronunciation of English (6th ed.).* Arnold Publishers, London, 2001.

[31] C. Cucchiarini and D. Binnenpoorte. Validation and improvement of automatic phonetic transcriptions. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, Denver, USA, 2002.

[32] P. Dalsgaard. Phoneme label alignment using acoustic-phonetic features and gaussian probability density functions. *Computer Speech and Language*, 6:303–329, 1992.

[33] L. Deng and D.X. Sun. A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features. *Journal of the Acoustical Society of America*, 95:2702–2719, 1994.

[34] S. Dharanipragada and S. Roukos. A fast vocabulary independent algorithm for spotting words in speech. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 98)*, Seattle, USA, 1998.

[35] V. Diakoloukas, V. Digalakis, L. Neumeyer, and J. Kaja. Development of dialect-specific speech recognizers using adaptation methods. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 97)*, Munich, Germany, 1997.

[36] R. Duda and P. Hart. *Pattern classification and scene analysis.* John Wiley and Sons, New York, USA, 1973.

[37] T. Dutoit. *An introduction to text-to-speech synthesis.* Kluwer, Dordrecht, The Netherlands, 1997.

[38] E. Eide. Automatic modeling of pronunciation variations. In *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech 99)*, Budapest, Hungary, 1999.

[39] E. Eide. Distinctive features for use in an automatic speech recognition system. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg, Denmark, 2001.

[40] J. Farinas and F. Pellegrino. Automatic rhythm modeling for language identification. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg, Denmark, 2001.

[41] M. Finke, J. Fritsch, D. Koll, and A. Waibel. Modeling and efficient decoding of large vocabulary conversational speech. In *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech 99)*, Budapest, Hungary, 1999.

[42] M. Finke and A. Waibel. Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech 97)*, Rhodes, Greece, 1997.

[43] E. Fosler-Lussier, I. Amdal, and H.-K.J. Kuo. On the road to improved lexical confusability metrics. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[44] E. Fosler-Lussier and N. Morgan. Effects of speaking rate and word frequency on pronunciations in conversational speech. *Speech Communication*, 29(2-4):137–158, 1999.

[45] J.E. Fosler-Lussier. *Dynamic pronunciation models for automatic speech recognition.* PhD thesis, University of California, Berkeley, 1999.

[46] J. Frankel and S. King. ASR - Articulatory Speech Recognition. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg, Denmark, 2001.

[47] O. Fujimura, S. Kiritani, and H. Ishida. Computer-controlled radiography for observation of movements of articulatory and other human organs. *Computers in Biology and Medicine*, 3:371–384, 1973.

[48] T. Fukada, T. Yoshimura, and Y. Sagisaka. Automatic generation of multiple pronunciations based on neural networks. *Speech Communication*, 27(1):63–73, 1999.

[49] M.J.F. Gales. The generation and use of regression class trees for MLLR adaptation. Technical Report CUED/F-INFENG/TR263, University of Cambridge, 1996.

[50] M.J.F. Gales. Cluster adaptive training for speech recognition. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP 98)*, Sydney, Australia, 1998.

[51] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren. The DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. Technical Report NISTIR 4930, National Institute of Standards and Technology, 1993.

[52] S. Gay, B. Lindblom, and J. Lubker. Production of bite-block vowels: acoustic equivalence by selective compensation. *Journal of the Acoustical Society of America*, 69:802–810, 1981.

[53] S.B. Gelfand, C.S. Ravishankar, and E.J. Delp. An iterative growing and pruning algorithm for classification tree design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):163–174, 1991.

[54] J.J. Godfrey, E.C. Holliman, and J. McDaniel. SWITCHBOARD: telephone speech corpus for research and development. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 92)*, San Francisco, USA, 1992.

[55] S. Goronzy. *Robust adaptation to non-native accents in automatic speech recognition.* Springer, Berlin, Germany, 2002.

[56] S. Greenberg. Speaking in shorthand - a syllable-centric perspective for understanding pronunciation variation. *Speech Communication*, 29(2-4):159–176, 1999.

[57] S. Greenberg. Whither speech technology? - a twenty-first century perspective. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg, Denmark, 2001.

[58] S. Greenberg and S. Chang. Linguistic dissection of Switchboard corpus automatic speech recognition systems. In *ISCA Tutorial and Research Workshop on Automatic Speech Recognition (ASR 2000)*, Paris, France, 2000.

[59] S. Greenberg, S. Chang, and J. Hollenback. An introduction to the diagnostic evaluation of Switchboard-corpus automatic speech recognition systems. In *NIST Speech Transcription Workshop*, College Park, USA, 2000.

[60] S. Greenberg. Personal communication.

[61] W.J. Hardcastle and N. Hewlett. *Coarticulation: theory, data and techniques.* Cambridge University Press, Cambridge, UK, 1999.

[62] J. Harris. *English sound structure.* Blackwell Publishing, Oxford, 1994.

[63] T.J. Hazen, I.L. Hetherington, H. Shu, and K. Livescu. Pronunciation modeling using a finite-state transducer representation. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[64] M.S. Hedrick and R.N. Ohde. Effect of relative amplitude on perception of frication place of articulation. *Journal of the Acoustical Society of America*, 94:2005–2026, 1993.

[65] T. Holter and T. Svendsen. Maximum likelihood modelling of pronunciation variation. *Speech Communication*, 29(2-4):177–191, 1999.

[66] C. Huang, E. Chang, J. Zhou, and K.-F. Lee. Accent modeling based on pronunciation dictionary adaptation for large vocabulary Mandarin speech recognition. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, 2000.

[67] J.J. Humphries. *Accent modelling and adaptation in automatic speech recognition.* PhD thesis, University of Cambridge, 1997.

[68] J.J. Humphries and P.C. Woodland. Using accent-specific pronunciation modelling for improved large vocabulary continuous speech recognition. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech 97)*, Rhodes, Greece, 1997.

[69] T. Imai, A. Ando, and E. Miyasaka. A new method for automatic generation of speaker-dependent phonological rules. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 95)*, Detroit, USA, 1995.

[70] The International Phonetic Association. URL: http://www.arts.gla.ac.uk/IPA/ipa.html.

[71] R. Jakobson, C.G.M. Fant, and M. Halle. *Preliminaries to speech analysis: the distinctive features and their correlates.* MIT Press, Cambridge, USA, 1952.

[72] D.A. James and S.J. Young. A fast lattice-based approach to vocabulary independent wordspotting. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 94)*, Adelaide, Australia, 1994.

[73] F. Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64:532–556, 1976.

[74] B.-H. Juang, W. Chou, and C.-H. Lee. Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 5:257–265, 1997.

[75] J.-C. Junqua and J.-P. Haton. *Robustness in automatic speech recognition: fundamentals and applications.* Kluwer Academic Publishers, Boston, USA, 1996.

[76] D. Jurafsky, W. Ward, Z. Jianping, K. Herold, Y. Xiuyang, and Z. Sen. What kind of pronunciation variation is hard for triphones to model? In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, Salt Lake City, USA, 2001.

[77] L.W. Kat and P. Fung. Fast accent identification and accented speech recognition. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 99)*, Phoenix, USA, 1999.

[78] J.M. Kessens and H. Strik. Lower WERs do not guarantee better transcriptions. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg, Denmark, 2001.

[79] J.M. Kessens, H. Strik, and C. Cucchiarini. Modeling pronunciation variation for ASR: comparing criteria for rule selection. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[80] J.M. Kessens, M. Wester, and H. Strik. Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation. *Speech Communication*, 29(2-4):193–207, 1999.

[81] S. King, T. Stephenson, S. Isard, P. Taylor, and A. Strachan. Speech recognition via phonetically featured syllables. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP 98)*, Sydney, Australia, 1998.

[82] S. King and P. Taylor. Detection of phonological features in continuous speech using neural networks. *Computer Speech and Language*, 14:333–353, 2000.

[83] K. Kirchhoff. Syllable-level desynchronisation of phonetic features for speech recognition. In *Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP 96)*, Philadelphia, USA, 1996.

[84] K. Kirchhoff. Combining articulatory and acoustic information for speech recognition in noisy and reverberant environments. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP 98)*, Sydney, Australia, 1998.

[85] K. Kirchhoff. *Robust speech recognition using articulatory information.* PhD thesis, University of Bielefeld, 1999.

[86] K. Kirchhoff. Integrating articulatory features into acoustic models for speech recognition. *Phonus 5*, pages 73–86, 2000.

[87] E. Kirshenbaum. Representing IPA phonetics in ASCII.
URL: http://www.kirshenbaum.net/IPA/ascii-ipa.pdf (unpublished), Hewlett-Packard Laboratories, 2001.

[88] T. Kohonen. *Self-organizing maps (3rd ed.).* Springer, New York, 2001.

[89] J. Koreman and B. Andreeva. Can we use the linguistic information in the signal ? *Phonus 5*, pages 47–58, 2000.

[90] J. Koreman, W.J. Barry, and B. Andreeva. Relational phonetic features for consonant identification in a hybrid ASR system. *Phonus 3*, pages 83–109, 1997.

[91] F. Korkmazskiy. Statistical learning of language pronunciation structure. In *ISCA Tutorial and Research Workshop on Automatic Speech Recognition and Understanding (ASRU 2001)*, Madonna di Campiglio, Italy, 2001.

[92] S. Koval, N. Smirnova, and M. Khitrov. Modelling pronunciation variability with hierarchical word networks. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[93] R. Kuhn, P. Nguyen, J.-C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini. Eigenvoices for speaker adaptation. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP 98)*, Sydney, Australia, 1998.

[94] K. Kumpf and R.W. King. Foreign speaker accent classification using phoneme-dependent accent discrimination models and comparisons with human perception benchmarks. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech 97)*, Rhodes, Greece, 1997.

[95] A. Lahiri. Speech recognition with phonological features. In *Proceedings of the 14th International Congress of Phonetic Sciences (ICPhS 99)*, San Francisco, USA, 1999.

[96] C.-H. Lee, C.-H. Lin, and B.-H. Juang. A study on speaker adaptation of the parameters of continuous density Hidden Markov Models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 39:806–814, 1991.

[97] K.-F. Lee and H.-W. Hon. Speaker-independent phone recognition using Hidden Markov Models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:1641–1648, 1989.

[98] K.-T. Lee, L. Melnar, and J. Talley. Symbolic speaker adaptation for pronunciation modeling. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[99] K.-T. Lee, L. Melnar, J. Talley, and C.J. Wellekens. Symbolic speaker adaptation with phone inventory expansion. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2003)*, Hong Kong, 2003.

[100] K.-T. Lee and C.J. Wellekens. Dynamic lexicon using phonetic features. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg, Denmark, 2001.

[101] K.-T. Lee and C.J. Wellekens. Dynamic sharings of Gaussian densities using phonetic features. In *ISCA Tutorial and Research Workshop on Automatic Speech Recognition and Understanding (ASRU 2001)*, Madonna di Campiglio, Italy, 2001.

[102] F. Lefèvre. *Estimation de probabilité non paramétrique pour la reconnaissance Markovienne de la parole*. PhD thesis, Université Pierre et Marie Curie, 2000.

[103] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density Hidden Markov Models. *Computer Speech and Language*, 9:171–185, 1995.

[104] D. McAllaster, L. Gillick, F. Scattone, and M. Newman. Fabricating conversational speech data with acoustic models: a program to examine model-data mismatch. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP 98)*, Sydney, Australia, 1998.

[105] H. Mokbel and D. Jouvet. Derivation of the optimal set of phonetic transcriptions for a word from its acoustic realizations. *Speech Communication*, 29(1):49–64, 1999.

[106] H. Ney. Architecture and search strategies for large-vocabulary continuous-speech recognition. In *Proceedings of the NATO ASI on New Advances and Trends in Speech Recognition and Coding*, Bubion, Spain, 1993.

[107] The NICO toolkit. URL: http://www.speech.kth.se/NICO.

[108] P. Nihalani, R.K. Tongue, and P. Hosali. *Indian and British English: a handbook of usage and pronunciation.* Oxford University Press, Delhi, 1979.

[109] M. Ostendorf, B. Byrne, M. Bacchiani, M. Finke, A. Gunawardana, K. Ross, S. Roweis, E. Shriberg, D. Talkin, A. Waibel, B. Wheatley, and T. Zeppenfeld. Modeling systematic variations in pronunciation via a language-dependent hidden speaking mode. In *Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP 96)*, Philadelphia, USA, 1996.

[110] G. Papcun, J. Hochberg, T.R. Thomas, F. Laroche, and J. Zacks. Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data. *Journal of the Acoustical Society of America*, 92:688–700, 1992.

[111] F. Pereira and M. Riley. Speech recognition by composition of weighted finite automata. In *Finite-State Language Processing (E. Roche and Y. Schabes, eds.)*, pages 431–453. MIT Press, Cambridge, USA, 1997.

[112] S.D. Peters and P. Stubley. Visualizing speech trajectories. In *ESCA Tutorial and Research Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, Rolduc, The Netherlands, 1998.

[113] P. Price. Combining linguistic with statistical methods in automatic speech understanding. In *Workshop on The Balancing Act*, Las Cruces, USA, 1994.

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

[115] M.C. Resnick. *Phonological variants and dialect identification in Latin American Spanish.* Mouton de Gruyter, The Hague, 1975.

[116] M. Richardson, J. Bilmes, and C. Diorio. Hidden-articulator Markov Models for speech recognition. In *ISCA Tutorial and Research Workshop on Automatic Speech Recognition (ASR 2000)*, Paris, France, 2000.

[117] M. Richardson, J. Bilmes, and C. Diorio. Hidden-articulator Markov Models: performance improvements and robustness to noise. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, 2000.

[118] M. Riley, W. Byrne, M. Finke, S. Khudanpur, A. Ljolje, J. McDonough, H. Nock, M. Saraclar, C. Wooters, and G. Zavaliagkos. Stochastic pronunciation modelling from hand-labelled phonetic corpora. *Speech Communication*, 29(2-4):209–224, 1999.

[119] M. Riley and A. Ljolje. Automatic generation of detailed pronunciation lexicons. In *Automatic Speech and Speaker Recognition: Advanced Topics*, pages 285–301. Kluwer, 1995.

[120] P. Roach and S. Arnfield. Variation information in pronunciation dictionaries. In *ESCA Tutorial and Research Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, Rolduc, The Netherlands, 1998.

[121] P. Roach and J. Hartman. *The English pronouncing dictionary (15th edition).* Cambridge University Press, Cambridge, UK, 1997.

[122] D.B. Roe and M.D. Riley. Prediction of word confusabilities for speech recognition. In *Proceedings of the 3rd International Conference on Spoken Language Processing (ICSLP 94)*, Yokohama, Japan, 1994.

[123] M. Saraçlar and S. Khudanpur. Pronunciation ambiguity vs pronunciation variability in speech recognition. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2000)*, Istanbul, Turkey, 2000.

[124] M. Saraçlar, H. Nock, and S. Khudanpur. Pronunciation modeling by sharing Gaussian densities across phonetic models. *Computer Speech and Language*, 14:137–160, 2000.

[125] P.W. Schönle, K. Gräbe, P. Wenig, J. Höhne, J. Schrader, and B. Conrad. Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract. *Brain and Language*, 31:26–35, 1987.

[126] S. Seneff and C. Wang. Modelling phonological rules through linguistic hierarchies. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[127] A. Sethy, S. Narayanan, and S. Parthasarthy. A syllable based approach for improved recognition of spoken names. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[128] T. Shinozaki and S. Furui. Error analysis using decision trees in spontaneous presentation speech recognition. In *ISCA Tutorial and Research Workshop on Automatic Speech Recognition and Understanding (ASRU 2001)*, Madonna di Campiglio, Italy, 2001.

[129] T. Sloboda. Dictionary learning: performance through consistency. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 95)*, Detroit, USA, 1995.

[130] T.A. Stephenson, H. Bourlard, S. Bengio, and A.C. Morris. Automatic speech recognition using dynamic Bayesian networks with both acoustic and articulatory variables. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, 2000.

[131] K.N. Stevens. On the quantal nature of speech. *Journal of Phonetics*, 17:3–45, 1989.

[132] K.N. Stevens. Applying phonetic knowledge to lexical access. In *Proceedings of the 4th European Conference on Speech Communication and Technology (Eurospeech 95)*, Madrid, Spain, 1995.

[133] I. Stobie. Uncertain future for speech recognition.
URL: http://www.vnunet.com/Analysis/1124939, 2001.

[134] A. Stolcke and S. Omohundro. Best-first model merging for Hidden Markov Model induction. Technical Report TR-94-003, International Computer Science Institute, 1994.

[135] H. Strik and C. Cucchiarini. Modeling pronunciation variation for ASR: a survey of the literature. *Speech Communication*, 29(2-4):225–246, 1999.

[136] J. Sun, X. Jing, and L. Deng. Data-driven model construction for continuous speech recognition using overlapping articulatory features. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, 2000.

[137] S. Suzuki, T. Okadome, and M. Honda. Determination of articulatory positions from speech acoustics by applying dynamic articulatory constraints. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP 98)*, Sydney, Australia, 1998.

[138] J. Tian, J. Häkkinen, and O. Viikki. Multilingual pronunciation modeling for improving multilingual speech recognition. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, Denver, USA, 2002.

[139] D. Torre, L. Villarrubia, L. Hernandez, and J.M. Elvira. Automatic alternative transcription generation and vocabulary selection for flexible word recognizers. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 97)*, Munich, Germany, 1997.

[140] M.-Y. Tsai, F.-C. Chou, and L.-S. Lee. Improved pronunciation modelling by inverse word frequency and pronunciation entropy. In *ISCA Tutorial and Research Workshop on Automatic Speech Recognition and Understanding (ASRU 2001)*, Madonna di Campiglio, Italy, 2001.

[141] V. Venkataramani and W. Byrne. MLLR adaptation techniques for pronunciation modeling. In *ISCA Tutorial and Research Workshop on Automatic Speech Recognition and Understanding (ASRU 2001)*, Madonna di Campiglio, Italy, 2001.

[142] R.A. Wagner and M.J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.

[143] Wagon (Edinburgh Speech Tools Library).
URL: http://www.cstr.ed.ac.uk/projects/speech_tools/.

[144] W. Ward, H. Krech, X. Yu, K. Herold, G. Figgs, A. Ikeno, D. Jurafsky, and W. Byrne. Lexicon adaptation for LVCSR: speaker idiosyncracies, non-native speakers, and pronunciation choice. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[145] C.J. Wellekens. *Traitement de la parole*. Institut Eurécom, Sophia Antipolis, France, 1998.

[146] J.R. Westbury, G. Turner, and J. Dembowski. *X-ray microbeam speech production database user's handbook*. Waisman Cetnter on Mental Retardation & Human Development, University of Wisconsin, Madison, USA, 1994.

[147] M. Wester and E. Fosler-Lussier. A comparison of data-derived and knowledge-based modeling of pronunciation variation. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, 2000.

[148] M. Wester, S. Greenberg, and S. Chang. A Dutch treatment of an elitist approach to articulatory-acoustic feature classification. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg, Denmark, 2001.

[149] M. Wester, J.M. Kessens, and H. Strik. Improving the performance of a Dutch CSR by modeling pronunciation variation. In *ESCA Tutorial and Research Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, Rolduc, The Netherlands, 1998.

[150] D. Willett, E. McDermott, and S. Katagiri. Unsupervised pronunciation adaptation for off-line transcription of Japanese lecture speeches. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[151] G. Williams and S. Renals. Confidence measures for evaluating pronunciation models. In *ESCA Tutorial and Research Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, Rolduc, The Netherlands, 1998.

[152] P.C. Woodland. Speaker adaptation: techniques and challenges. In *ISCA Tutorial and Research Workshop on Automatic Speech Recognition and Understanding (ASRU 99)*, Keystone, USA, 1999.

[153] M. Woszczyna. *Fast speaker independent large vocabulary continuous speech recognition.* PhD thesis, University of Karlsruhe, 1998.

[154] A. Wrench. A multichannel/multispeaker articulatory database for continuous speech recognition research. In *Phonus 5*, Saarbrücken, Germany, 2000.

[155] Q. Yang and J.-P. Martens. Data-driven lexical modeling of pronunciation variations for ASR. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, 2000.

[156] Q. Yang, J.-P. Martens, P.-J. Ghesquiere, and D. Van Compernolle. Pronunciation variation modeling for ASR: large improvements are possible but small ones are likely to achieve. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

[157] L. Yi and P. Fung. Model partial pronunciation variations for spontaneous Mandarin speech recognition. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, Denver, USA, 2002.

[158] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK book, version 2.2.* Cambridge University Engineering Department, Cambridge, UK, 1999.

[159] S.J. Young, N.H. Russell, and J.H.S. Thornton. Token passing: a simple conceptual model for connected speech recognition systems. Technical Report CUED/F-INFENG/TR.38, Cambridge University Engineering Department, 1989.

[160] S.J. Young and P.C. Woodland. State clustering in Hidden Markov Model-based continuous speech recognition. *Computer Speech and Language*, 8:369–383, 1994.

[161] J. Zacks and T.R. Thomas. A new neural network for articulatory speech recognition and its application to vowel identification. *Computer Speech and Language*, 8:189–209, 1994.

[162] A. Zierdt, P. Hoole, and H.G. Tillmann. Development of a system for three-dimensional fleshpoint measurement of speech movements. In *Proceedings of the 14th International Congress of Phonetic Sciences (ICPhS 99)*, San Francisco, USA, 1999.

[163] G.G. Zweig. *Speech recognition with dynamic Bayesian networks.* PhD thesis, University of California, 1998.

Index

Curriculum Vitae

Name:	Kyung-Tak Lee
Date of Birth:	17 September 1974
Place of Birth:	Seoul, Republic of Korea

Education:

1999 - 2003	**EPFL, Swiss Federal Institute of Technology (Lausanne, Switzerland) & Eurécom Institute (Sophia-Antipolis, France)** Ph.D. thesis on "Dynamic Pronunciation Modeling using Phonetic Features and Symbolic Speaker Adaptation for Automatic Speech Recognition"
1999 - 2000	**University of Nice Sophia-Antipolis (France)** M.Sc. in image processing and computer vision – with distinction
1994 - 1999	**EPFL, Swiss Federal Institute of Technology (Lausanne, Switzerland) & Eurécom Institute (Sophia-Antipolis, France)** M.Sc. in telecommunications – emphasis in multimedia communications
1990 - 1994	**Candolle High School (Geneva, Switzerland)** Maturity certificate – highest honors, awards in Mathematics, Physics and German

Work Experience:

2002	**Motorola (Schaumburg, IL, United States)** Research on symbolic speaker adaptation for automatic speech recognition
1999 - 2001	**Eurécom Institute (Sophia-Antipolis, France)** Research on pronunciation variation modeling for automatic speech recognition Supervision of students' projects and practical works
Jan - Jun 1999	**Motorola (Palo Alto, CA, United States)** Research on handwriting recognition – created a prototype equation editor
Sep - Dec 1998	**Eurécom Institute (Sophia-Antipolis, France)** Multimedia – created a prototype Web server that provides personalized video news
Jul - Oct 1997	**Siemens (Lausanne, Switzerland)** Telecommunications – created usability tests for a traffic network management program
1996 - 1998	**EPFL, Swiss Federal Institute of Technology (Lausanne, Switzerland)** Telecommunications – created a program that better manages traffic networks Computer science – created an user interface for the "Mathematica" tool Computer science – evaluated performance of a compression layer over TCP/IP
Oct 1995	**Samsung / Telecom95 (Geneva, Switzerland)** Telecommunications – interpreter and presenter of the CDMA system from Samsung

Publications:

K.-T. Lee, L. Melnar, J. Talley and C.J. Wellekens. **Symbolic Speaker Adaptation with Phone Inventory Expansion**. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2003)*, Hong Kong, 2003.

K.-T. Lee, L. Melnar and J. Talley. **Symbolic Speaker Adaptation for Pronunciation Modeling**. In *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA 2002)*, Estes Park, USA, 2002.

K.-T. Lee and C.J. Wellekens. **Dynamic Sharings of Gaussian Densities using Phonetic Features**. In *ISCA Tutorial and Research Workshop on Automatic Speech Recognition and Understanding (ASRU 2001)*, Madonna di Campiglio, Italy, 2001.

K.-T. Lee and C.J. Wellekens. **Dynamic Lexicon using Phonetic Features**. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg, Denmark, 2001.

B. Merialdo, K.-T. Lee, D. Luparello and J. Roudaire. **Automatic Construction of Personalized TV News Programs**. In *Proceedings of the 7th ACM International Multimedia Conference (ACM MM'99)*, Orlando, USA, 1999.