Explicit N–Best Formant Features for Segment–Based Speech Recognition

Philipp Schmid
Lizentiat, University of Bern, Switzerland, 1990

A dissertation submitted to the faculty of the
Oregon Graduate Institute of Science & Technology
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Computer Science and Engineering

October 1996

The dissertation "Explicit N–Best Formant Features for Segment–Based Speech Recognition" by Philipp Schmid has been examined and approved by the following Examination Committee:

Dr. Etienne Barnard
Associate Professor
Thesis Research Adviser

Dr. Mark Fanty
Associate Professor

Dr. Ronald Cole
Professor

Dr. Victor Zue
Professor
Massachussetts Institute of Technology

Dedication

To Julie and my parents, Grete and Heinz.

Acknowledgements

At the end of a long journey I have a lot of people to thank. First and foremost, I would like to thank Ron Cole for bringing me to OGI and for giving me the opportunity to pursue my ideas, to travel the world, and to interact with my peers and his peers through five long and very rewarding years. I will never forget his first words to me at OGI: "Are you aware that we will send you home if you don't pass the exam? Oh, by the way, welcome to OGI".

Next, my gratitude goes to my two thesis advisors over the years: Mark Fanty and Etienne Barnard. Mark supported me in my attempts to build a segment–based framework and Etienne pointed me to the idea of N–best formant tracking. I want to thank them for showing me how to do proper research and for teaching me the finer details of statistical analysis and stochastic–classifier design. Additionally, I would like to thank Victor Zue for agreeing to serve on my committee and for taking his duties as the outside advisor very seriously during my defense. My thanks also go to Jim Glass, Beatrice Oshika and Jim Hieronymus for taking the time to listen to my ideas and for giving me feedback and encouragement.

I will always remember my fellow students at OGI, starting with Jenny and the Gang of Eleven for helping me to study for the quals. It was one of my greatest experiences at OGI. And then there are my office mates, Karen and Kay, who had to put up with my constant chess playing against the computer. A special thanks goes to Karen and Dan for helping me prepare for presentations, as well as Don and Hong for many stimulating discussions, and to Johan, Sarel and Jacques for helping me with signal processing and programming problems.

I have been fortunate enough to have had two wonderful roommates over the years, Ron and Takayuki. It was a pleasure sharing an apartment with them in a beautiful NW

Portland neighborhood. Also, the travels with Kuno and Julie are as much part of my Portland experience, as the chess playing with Steve in his bookstore on NW 23rd.

I also want to thank my parents for supporting me way back in 1981 when I first approached them with the crazy idea of going to America to attend high school, and for encouraging me to pursue my dreams. Their frequent visits to Portland helped me get over the sporadic feelings of homesickness.

And finally, my eternal gratitude goes out to Julie for always being there when I needed her support and for cheering me up and encouraging me when I felt down. I would not have been able to achieve everything I did without her.

Contents

Dedication . . . . . **iii**

Acknowledgements . . . . . **iv**

Abstract . . . . . **xi**

1 Introduction **1**
- 1.1 Spectrogram Reading 3
- 1.2 Segmental Framework 6
- 1.3 N–Best Formant Features 8
- 1.4 Overview 11

2 Related Work **14**
- 2.1 Segment–based Speech Recognition Systems 14
  - 2.1.1 Segmentation Methods 15
  - 2.1.2 Segment Classification 19
  - 2.1.3 Search Algorithms 21
- 2.2 Formant Tracking Algorithms 24
  - 2.2.1 ESPS Formant Tracker 25
  - 2.2.2 CRIN / INRIA 2–Pass Formant Tracker 26
- 2.3 Perceptual / Formant Theory 27
- 2.4 Segment Classification Experiments 28
- 2.5 Phonetic Recognition Experiments 30

3 Segmentation **32**
- 3.1 Acoustic Segmentation 34
- 3.2 Evaluation of Boundary Detection 36
- 3.3 Acoustic Segment Classification 37
- 3.4 Construction of the Segment Lattice 38
- 3.5 Evaluation 41
- 3.6 Summary 43

4 Segment Classification **44**
4.1 Data Sets and Signal Processing 45
4.2 Baseline 46
4.3 Machine Segments 47
4.4 Training on Errors 49
4.5 Summary 51

5 Robust, N–best Formant Tracking **52**
5.1 Finding Elementary Tracks 53
5.2 N–best Search 55
5.3 Evaluation 58
5.4 Summary 60

6 Explicit Formant Features **61**
6.1 Knowledge–Based Features 61
6.2 Piecewise–Linear Regression 63
6.3 Legendre Polynomials 66
6.4 N–best Classification Paradigm 68
6.5 Summary 72

7 Phonetic Recognition **73**
7.1 Probabilistic Framework 73
7.2 Implementation Issues 75
7.2.1 Modified Boundary Probabilities 76
7.2.2 Augmented Segment Lattice 78
7.3 Recognition Results 78
7.4 Summary 81

8 Conclusions and Future Work **83**
8.1 Thesis Summary and Contributions 83
8.2 Future Work 86
8.2.1 Porting to Telephone–Quality Speech 87
8.2.2 Extensions to the Basic Paradigm 89
8.3 Famous Last Words 92

Bibliography **93**

A Distance Functions **101**
A.1 Overview 101
A.2 Context–Dependent Spectral Variation Function 102
A.3 Discussion 103

Biographical Note **105**

List of Tables

1.1 Glossary of Mathematical Symbols 13

2.1 Overview of the Architectural Features 15
2.2 Overview: Vowel Classification Results 29
2.3 Overview: Phonetic Segment Classification Results 30
2.4 Overview: Phonetic Recognition Results 31

3.1 Acoustic Segment Boundary Detection Results 37
3.2 Lattice Generation Rules 40
3.3 Alignment Statistics 42

4.1 Training and Test Set Statistics 45
4.2 Phoneme Set 46
4.3 Classification Results for Baseline Classifiers 47
4.4 Classification Results using Machine Segmentations 50

5.1 Diversity of the N–best Formant Tracker 59

6.1 Vowel Classification Results For PLR Features 65
6.2 Legendre Polynomials 67
6.3 Vowel Classification Results For LEG Feature Sets 67
6.4 Machine–Segmentation Results 68
6.5 Classification Results for Iterative Training Procedure 70
6.6 Classification Results for Oracle Experiment 71

7.1 Types of Recognition Errors 79
7.2 Definitions of Performance Measurements 79
7.3 Phonetic Recognition Results 80

8.1 Thesis Contributions 86
8.2 Porting Steps 87

List of Figures

1.1 Explain creation of a spectrogram 3
1.2 Spectrogram Reading Example 5
1.3 N–best Idea 9

3.1 Example of "slow" (/r/–/iy/) and "fast" (/s/–/iy/) boundaries 33
3.2 Boundary Detection Scoring Process 36
3.3 Acoustic segment classification features. 38
3.4 Lattice Generation Rules 39

4.1 Features for the Baseline Segment Classifier 46
4.2 Positive and negative training segments 49

5.1 Elementary Track Finding Algorithm 54
5.2 N–best Search Algorithm 55
5.3 Initialization Rules for the Search 56
5.4 Credit Assignment Rules 56
5.5 Expansion Rules for the Search 57
5.6 Formant Tracking Example 58
5.7 Top 3 Formant Interpretations 59

6.1 Piecewise Linear Regression Algorithm 63
6.2 PLR Formant Features 64
6.3 PLR Formant Features 65
6.4 Iterative Training Procedure for Sonorant Segment Classifier 69

7.1 Search Hypotheses 77
7.2 Augmentation of Segment Lattice due to single Segment D 78

A.1 MLP for computing the context-sensitive weighting coefficients 102
A.2 CD–SVF Example 104

Abstract

Explicit N–Best Formant Features for Segment–Based Speech Recognition

Philipp Schmid, Ph.D.
Oregon Graduate Institute of Science & Technology, 1996

Supervising Professor: Dr. Etienne Barnard

This thesis investigates the use of explicit speech knowledge in computer speech–recognition. Speech knowledge is generally expressed in terms of acoustic events occurring near phonetic segment boundaries and the location, shape and dynamics of formant trajectories. This suggests the creation of a segment–based recognition framework and the use of explicit formant features in a flexible integration scheme to ultimately improve the phonetic recognition accuracy.

We describe a segmentation algorithm that produces a lattice of segment hypotheses, each with an associated broad phonetic identity. We build a single phonetic segment classifier along with separate vowel/semi–vowel and consonant classifiers based on traditional cepstral features paying attention to reducing the mismatch between training and deployment conditions.

We develop a robust, N–best formant tracking algorithm that generates a list of up to N consistent formant interpretations. The use of the N–best feature paradigm is based on the observation that there are generally only a handful of reasonable interpretation of the given formant information. Instead of finding the best formant interpretation through the use of a global cost function that includes energy maximization and smoothness terms,

we delay the selection of the correct formant interpretation until after the segment classification and phonetic search.

We use the formant interpretations to extract features for a vowel/semi–vowel segment classifier. The formant trajectories are approximated either by three line segments or by a third–order Legendre polynomial. We show that together with formant amplitude, formant bandwidth, pitch, and segment durations we can produce a classifier of comparable performance to a cepstral–based classifier. We further demonstrate the potential of the N–best classification paradigm and show that a combination of formant and cepstral features further improves the classification accuracy. Finally, the validity of the entire approach of using a segment–based approach, separate classifiers for vowels and consonants, and explicit formant features is verified by phonetic recognition experiments.

Chapter 1

Introduction

Spoken language is the preferred mode of communication among humans, for a variety of reasons [71]. Therefore it was only natural for researchers to investigate the use of spoken natural language as a way of communicating with machines to access information or to issue commands: the field of automatic speech recognition (ASR) was born.

Numerous real–world deployments of speech–recognition systems are starting to gain public acceptance: cellular voice dialers (often used in automobiles), voice routing of calls, automatic transcription of medical reports, order entries at insurance companies, and limited voice dictation are among the most popular ones.

The speech–recognition application used by most people at this point is the automatic billing function of certain operator services used by AT&T. An estimated 4 billion calls per year were handled by machines in 1993 [56]. A study by Voice Information Associates [87] determined that the end–user market for ASR in 1994 totaled approximately 350 million dollars and projects the sales to increase to 1 billion dollars by the year 1998.

Despite this impressive list of accomplishments we should not loose sight of the fact that these systems' capabilities are a long way away from the science–fiction machines with speech–recognition capabilities such as *Hal* in Kubrik's "2001, A Space Odyssey" or the android *Data* on the TV series "Star Trek the Next Generation". The performance of a computer speech recognizer (possibly in conjunction with gesture recognition) will have to be near human capabilities before it is widely accepted. Because ASR generally does not augment our daily lives with capabilities outside our natural abilities (with some exceptions such as paraplegics and typists suffering from carpal–tunnel syndrome) but rather replaces something that humans are very proficient at and therefore the machine's

performance will be measured against it. Contrast this with the computation of the trajectory of the moon rocket which is generally beyond the capabilities of humans but which the computer can do in the blink of an eye.

As more and more research systems are deployed in real–world environments, we realize that they are fragile. These initial systems are lacking in basic recognition accuracy, both at the phonetic and the word level, in comparison to human standards [58]. Additionally, the recognition performance degrades considerably under adverse conditions, such as the presence of background noise.

Researchers have been able to compensate to some extent for these shortcomings by taking advantage of improved contextual constraints (language models, dialog models). However, we cannot overlook our lack of scientific understanding of those accuracy and robustness problems. As the field is moving towards recognizing spontaneous speech, we will not be able to rely on grammatical constraints to the degree that the current systems do if we want to obtain a reasonable level of naturalness. With the use of large vocabularies (10,000 words or more) the acoustic distance between the words decreases and hence the confusability increases. The only way to remedy these problems is to improve the basic phonetic recognition accuracy of these systems and to increase their robustness to background noise. This thesis represents the begining of a line of research focusing on improving the phonetic recognition accuracy of automatic speech recognition systems.

In the absence of syntactic and semantic constraints, expert spectrogram readers achieve more than 90% correct identification of phonetic segments [17, 20, 92], whereas today's best laboratory systems achieve only approximately 70% accurate recognition on the same task [31, 59]. What is the reason for this big difference in accuracy between experts and automatic speech–recognition systems? It is not just the increased accuracy of the segmentation done by the human expert, because the segment classification results (when the correct segmentation is known to the classifier) are below 80% [14, 52]. Therefore, it can be argued that one of the major differences between the two is the use of explicit speech knowledge by the experts to segment the speech and to classify the phonetic segments. This suggests that one way to improve the accuracy and robustness of current systems is to incorporate explicit speech knowledge, of the type used by expert spectrogram readers,

into the recognition algorithms. Ideally, we would like to model the biological recognition system, but unfortunately our understanding thereof has not progressed to a point that would allow us to implement the same principles in computer algorithms. Therefore, our research will focus on the principles of spectrogram reading, some of which are described in the next section.

1.1 Spectrogram Reading

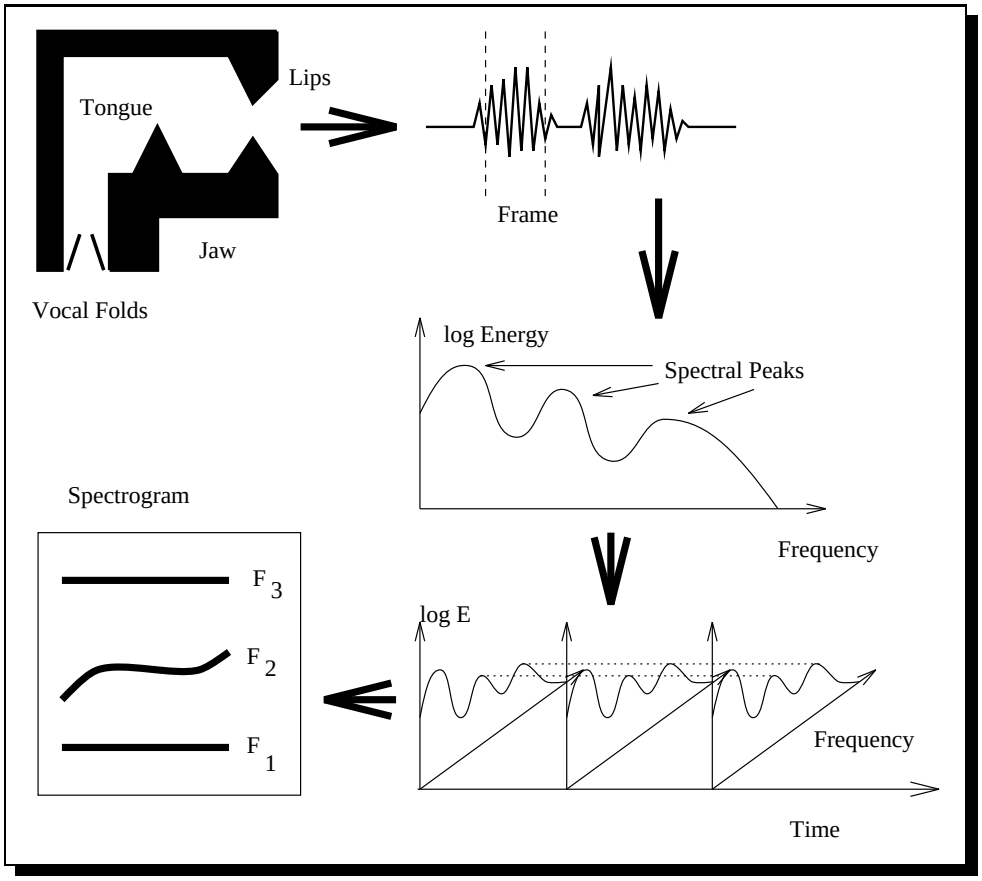

Figure 1.1: This figure explains the creation of speech spectrogram [from top left]: speech production model, speech waveform, logarithmic power spectrum, multiple spectra over time, and finally a 2-D projection.

Before we describe the process of spectrogram reading, we briefly explain the construction of a spectrogram while at the same time introducing some important terms that will

be used throughout this thesis. Figure 1.1 illustrates schematically the contruction of a speech spectrogram, starting with the speech production model in the upper left corner. Different articulators, such as vocal folds, tongue, jaw, and lips influence the exitation and resonance characteristics of the oral and nasal cavities. Depending on the type of activation (voiced/unvoiced) and positioning of the articulators different speech sounds, called phonemes, can be produced (see [10, 43] for details on speech production). Phonemes are the basic units of speech used to construct words and meaning. There are approximately 40 phonemes in the English language. This basic phoneme set can be further subdivided based on the manner of articlation into vowels, semi–vowels (liquids and glides), nasals, fricatives, stops, and closures.

Information regarding the state of the vocal tract can be captured by recording the vibrations of the exiting air flow using a microphone. The resulting digital speech waveform is generally spliced into 5 or 10 millisecond (ms) frames. Next, the frequency composition of each frame is analyzed using Fourier analysis [74] and plotted in a frequency–(logarithmic) energy plane called a spectrum. For vowel–like sounds, the spectrum will generally show several spectral peaks indicating the resonant frequencies of the vocal tract. Typically, there are 3 peaks in the range between 300Hz and 4kHz, although in practice (as we will discuss later) the number of peaks may vary. Spectra from all speech frames are combined into a 3D–surface display which is finally projected into the time–frequency plane encoding the energy as an intensity value. The resulting spectrogram has dark bands (stemming from spectral peaks, which in turn are related to resonances in the vocal tract) called formants. The three formants of lowest frequency are numbered as F1, F2 and F3. The formants play a central role in spectrogram reading, as discussed next.

We will illustrate the process of spectrogram reading using Figure 1.2. First, the utterance is segmented into acoustically similar regions, called segments (indicated by vertical dashed lines in Figure 1.2). Next, a quick assessment as to the broad–phonetic nature of the segment (generally based on the manner of articulation and voicing categorization) is performed. Finally, a fine–phonetic analysis is performed to determine the identity of the phonetic segment. Consonants are uniquely described in terms of (1) voicing, (2) place of articulation, and (3) manner of articulation. Vowels are generally described in terms of

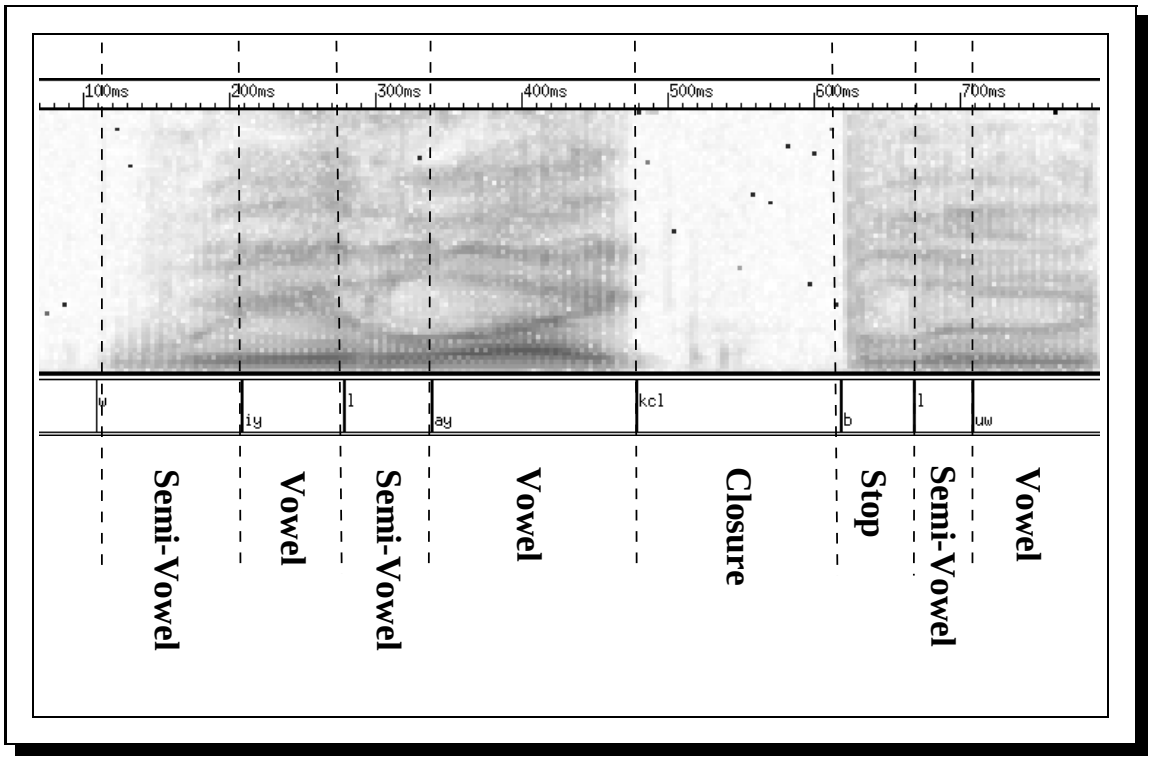

Figure 1.2: Spectrogram Reading Example of the utterance "We like". The phonetic segmentation is indicated by vertical dashed lines. The place–of–articulation broad–phonetic classification is shown below the spectrogram for each segment.

(1) the height of the body of the tongue, (2) the front–back position of the tongue, and (3) the degree of lip rounding [45]. Formants are the primary indicator for the place of articulation. Additionally, they play a central role in determining the characteristic vowel features mentioned above. This fact motivates our interest in formants in this thesis.

If we want to mimick the principles of spectrogram reading we need to employ a segment–then–classify approach to speech recognition. This approach is best characterized by Zue et al. [93]:

> "Therefore, one must explicitly establish acoustic landmarks in the speech signal in order to fully utilize these acoustic attributes. Second, ..., we seek to make use of the available speech knowledge by embedding such knowledge into a formal framework whereby powerful mathematical tools can be utilized to optimize its use. Third, the system must have a stochastic component to deal with the present state of ignorance in our understanding of the human communication process and its inherent variability throughout".

This research also uses a a segment–and–classify approach to automatic speech recognition. This approach has been successfully applied to speech recognition and spoken language systems as can be seen from our overview (Chapter 2: "Related Work"). We will use the term *segment–based* or *segmental* system framework to refer to this general concept. In the next two sections we discuss the major design decisions underlying this thesis research: the segment–based recognition framework and the use of explicit formant features in an N–best paradigm.

1.2 Segment–Based System Framework

A segment–based recognition framework has the following potential advantages over more traditional frame–based approaches:

- Explicit segmentation allows us to apply acoustic–phonetic knowledge, gained from statistical analyses, spectrogram reading, and perceptual experiments, directly to the classification task. For example, when classifying stops, the duration of the consonant, the presence or absence of voicing in the preceding closure and the formant transitions at the onset of the following vowel can all be estimated. Explicit segmentation allows us to locate and measure those features of the signal that are important for perception (and hence possibly useful for the classification as well).

- The correlation between successive frames within a phonetic segment can be modelled and used as a powerful classification feature. The Markovian independence assumption used by the Hidden Markov models (HMM) (see [73] for a tutorial on HMMs) is generally not valid. Hence modelling the state dynamics and statistical dependencies is more realistic and will lead to more accurate probability estimates. In contrast, HMMs model the speech signal as a sequence of statistically independent frames and thus underestimate the likelihood of the sequence[1].

- Separate classifiers can be used for different broad phonetic classes, such as different classifiers for sonorant and obstruent segments. (However, the proper combination their probability estimates might be difficult in practice.)

  The set of feature measurements can be designed and optimized for individual discriminations (when using different segment classifiers). The important information about the nature of a vowel segment for example is contained within the phonetic segment, whereas nasals can be distinguished primarily by the formant transitions into and out of the nasal segment rather than by the spectral shape within the segment.

- Segmental features such as segment duration, average pitch, energy, and zero crossing measurements can be defined and extracted in a meaningful manner.

- Because features and classifiers are optimized for individual discriminations, classification errors can be analyzed and improved with minimal effect on other discriminations.

At this point we should not overlook the disadvantages of an explicit segmentation step, such as the loss of flexibility once a segmentation has been derived. Once the segmentation has been fixed, mismatches between expected pronunciations and possible search results can cause recognition errors (e.g. the correct word is a sequence of k phonemes where as no reasonable sequence through the segment lattice is of length k). Frame–based systems

[1] The likelihoods are underestimated since systematic differences from the mean values are penalized at each frame.

are more flexible in dealing with inaccurate pronunciation modeling since the number of frames (segments) per word is not fixed. Additionally, as we will discuss in Chapter 7, the definition of the probability of a given segmentation being correct or acoustically plausible (as used in the probabilistic framework for most segment–based systems described in Section 2.1.3) is difficult and largely an unsolved problem at this point.

Apart from the segmental features, the most beneficial aspect of this segment–based approach is the opportunity to model the dynamics of speech within a phonetic segment in a more principled manner as discussed in the next section.

1.3 Formant Features in an N–best Classification Paradigm

The above arguments on the importance of formants in spectrogram reading suggest the use of formants as the primary feature representation for speech recognition. This idea was popular in the late 1970's as part of the ARPA speech recognition program [39]. As a matter of fact, some of the main ideas in this thesis, multi–level segmentation (initially introduced by Wolf et al. [90]), using multiple, specialized classifiers, and the use of formant features for sonorant classification were proposed by Weinstein et al. [88]. However, the accuracy and consistency of the formant trackers used were below the levels needed for speech recognition. Making a final decision (with regard to formant locations) early in the recognition process proved to be an obstacle difficult to overcome. Nevertheless, formant features were successfully used in the FEATURE system [18] to classify the letters of the English alphabet.

Additionally, perceptual experiments conducted by Klatt [40] showed that formant frequency information are more important for vowel perception as compared to other factors such as spectral tilt, relative formant amplitudes, high and low–pass filtering).

In his description of the state–of–the–art formant tracker (for a summary see Section 2.2.1), Talkin [83] discusses the various options in implementing a formant tracker and mentions three particular problems with frame–based formant tracking algorithms:

1. The identification of formant candidates for each frame: what is the algorithm to do if the number of candidates differs from the number of formants to be tracked?

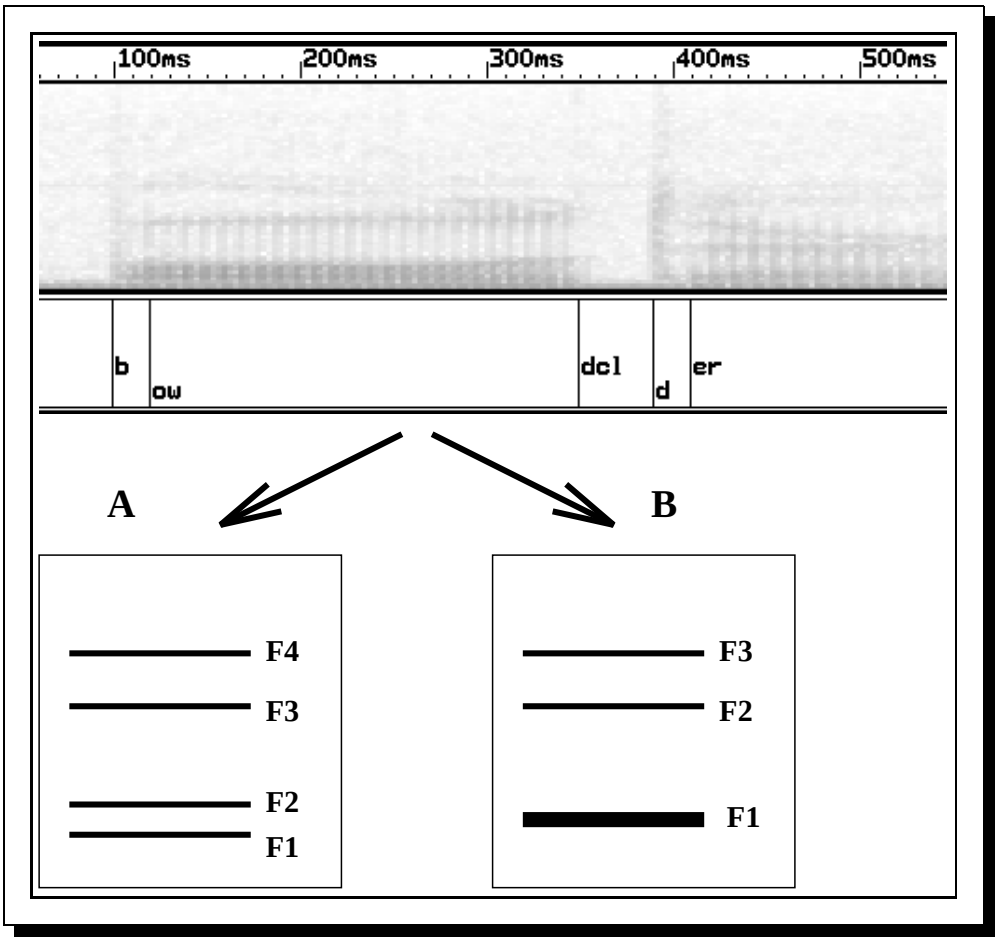

Figure 1.3: Motivation of the N–best paradigm. The first vowel /ow/ of the word "boulder" has a a broad first formant, which can be interpreted in one of two ways: either two formants are close together and couldn't be separated by the visual representation (A), or nasalization of the vowel is broadening the formant (B).

2. The enforcement of smoothness constraints across vowel/consonant boundaries: tight constraints are desirable within sonorant segments but cause problems in unvoiced segments since there are no real formant peaks to follow.

3. The trade–off problem: formant trajectory smoothness versus maximizing the amount of spectral energy explained by the chosen formant locations.

Chapter 5 describes our implementation of a robust, N–best formant tracker that attempts to address the problems outlined above. Firstly, it uses an N–best tracking paradigm: instead of finding the single best interpretation given an objective function that incorporates energy and smoothness goals, this algorithm finds the N most consistent interpretations. In this context, consistency is referred to as desirable behavior, such as following energy concentrations, producing a smooth trajectory, and avoiding large jumps from one frame to the next. This addresses problem 3 from above. The N–best idea is illustrated in Figure 1.3. The broadening of the lowest formant can either be the result of two close formants which the visual representation cannot separate or be the result of nasalization which tends to increase the bandwidth of the first formant. In our N–best paradigm, both possibilities, (A) and (B), would be passed on to the segment–classification stage for further consideration, where as in the traditional formant–tracking paradigm only one of the two solutions will be considered further.

The flexibility of our tracking algorithm is further improved by using wild cards in the search for the best interpretations. Wild cards are inserted into the search in instances where no logical extension of a hypothesis is possible (see Figure 5.6 for an example of using a wild card). This allows us to handle cases where fewer than three formant candidates[2] were detected by the preprocessing stage (problem 1). And finally, we will apply the formant tracker only to sonorant regions (as determined by the lattice generation process), thus simplifying problem 2 to the proper identification of sonorant speech.

Separate features are extracted for each of the N interpretations found by the formant tracking algorithm, which are then presented to a vowel/semi–vowel classifier (see Chapter

[2]In this research, we are only interested in F1, F2, and F3, as they are the primary carrier of information for vowel classification. The algorithm however can easily be extended to track more than three formants if desired, e.g. for consonant classification.

6). The selection of the correct interpretation and phonetic category is thus delayed until after the classification stage (or alternatively, in the case of word recognition, until after the word search) where the interpretation and phonetic category with the highest posterior probability (in the case of phonetic recognition) is selected. Note that we use different features for the same segment in this process! The problem will be the comparison of the individual probability estimates.

The use of formant features together with this N–best classification approach (described in more detail in Chapter 6) has several potential advantages over more conventional classification features (Chapter 4). For instance, as has been pointed out by Allen [1], noise in a particular frequency band influences all cepstral and spectral coefficients. On the other hand, a formant representation is more robust to such noise. Either the noise is in a frequency band not occupied by a formant in which case no distortion is observed, or the estimation of the formant is obscured, but will be recovered by using consistency constraints with respect to the adjacent frames to estimate the formant location. The same process will reduce the problems for heavily glottalized speech as well as strong extraneous noise events such as clicks. Similarly, principled speaker normalization should be facilitated when formant frequencies are known explicitly [65].

The Mel–frequency cepstral coefficients [64], commenly used in today's speech recognition systems, have the advantage that there is no tracking necessary. However, the salient information (with regards to spectral shape) is spread over multiple coefficients. Additionally, cepstral coefficients form complex featrure trajectories over time [36], requiring sophisticated modeling techniques (see Chapter 2 for a summary).

1.4 Overview of the Thesis

The goal of this thesis research is to create a flexible recognition framework for the explicit incorporation of speech knowledge into speech–recognition algorithms. As a demonstration of the validity of this approach, we will explicitly model the dynamics of speech through formant tracking in an N–best feature paradigm.

In the next chapter we summarize previously published research related to the topics

covered in this thesis: segment–based systems, perceptual experiments, and formant–tracking algorithms. We have argued above for a segment–based recognition framework. Chapter 3 describes the segmentation algorithm, similar to the dendrogram [28], which generates a lattice of segment hypotheses each with an associated broad category identity. Initial segment classifiers (phones, vowels, and consonants) are constructed in Chapter 4. We touch on some training issues and show that the classification performance can be improved by training the classifiers on hand–labeled as well as machine–segmented segments.

Next, in Chapter 5, we describe our robust, N–best formant tracker which produces a list of "consistent" formant interpretations. We experiment with different methods to approximate the formant trajectories, and add additional features, such as pitch and formant amplitude, to construct a vowel/semi–vowel segment classifier, described in Chapter 6. We present an iterative training algorithm to compensate for the lack of labeled training data and show the potential of the N–best classification scheme.

The different classifiers (based on either cepstral or formant features) are combined in phonetic recognition experiments described in Chapter 7, where we also address the problems in deriving a probabilistic framework for the segment–based recognition. Finally, Chapter 8 summarizes the findings of this research and outlines future work to extend these initial findings.

The publically available TIMIT acoustic–phonetic speech corpus [46] is used for the experiments throughout this thesis. Additionally, the spectrogram displays were generated using selected utterances from the TIMIT corpus.

The items in the glossary (Table 1.1) are used throughout this thesis, and are provided as a reference here.

Table 1.1: Glossary of Mathematical Symbols

Glossary of Mathematical Symbols

A	$=$	$\{a_1, a_2, \ldots, a_N\}$	Phonetic String
a_i	$\in$	$\{/ah/, /eh/, \ldots\}$	Phonetic Labels
β_i	$\in$	$\{/closure/, /vowel/, \ldots\}$	Broad–Phonetic Labels
X	$=$	$\{x_1, x_2, \ldots, x_T\}$	Acoustic Vectors (T Frames)
$\mathcal{AS}$	$=$	$\{as_1, as_2, \ldots, as_K\}$	Acoustic Segmentation
as_k	$=$	$\{x_l, x_{l+1}, \ldots, x_m\}$	Acoustic Segment from Frame l to Frame m
b_k			Boundary between as_k and as_{k+1}
$\mathcal{S}$	$=$	$\{S_1, S_2, \ldots, S_M\}$	Set of Segmentations
S_i	$=$	$\{s_0, s_1, \ldots, s_M\}$	Segmentation
s_i	$=$	$\{as_m, as_{m+1}, \ldots, as_n\}$	Segment

Chapter 2

Related Work: Segment–based Speech Recognition Systems and Formant Tracking

There are two major aspects to this research: the segment–based recognition framework and the explicit, N–best formant features for segment classification. In the first part of this chapter, we discuss various segment–based speech recognition systems reported in the literature. A commercially available formant tracker and the work most closely related to our own formant tracking algorithm are described in the second part, followed by a brief summary of perceptual experiments of vowel recognition. Finally, we summarize the phonetic classification and recognition results reported in the literature.

2.1 Segment–based Speech Recognition Systems

Segment–based systems can be characterized by three architectural features: segmentation method, segment classification technique, and search algorithm. We discuss these three components separately using prototypical systems reported in the literature. These systems span a wide range of solutions: from the single segmentation and reclassifying OGI Alphabet Recognizer [16] to the Stochastic Segment Modelling (SSM) system proposed by Ostendorf and Roukos [68], which considers every possible segmentation. Other systems represent a compromise between these two ends of the spectrum.

In the following we are contrasting systems with respect to the three architectural features. Table 2.1 summarizes the main features for each of the systems discussed below.

In this review of related work, we are not addressing issues related to the classification and recognition of context–dependent units (e.g., tri–phones), since we do not perform any context–dependent experiments.

Table 2.1: Overview of the Architectural Features for various Segment–based Systems. fsync = Frame–synchronous Viterbi search, bsync = Boundary–synchronous Viterbi search.

Architectural Features

System	Segmentation	Classification Features	Search
SUMMIT [94]	Dendrogram	Acoustic Attributes	A*
Tracks [31]	Implicit	Track Models	fsync
SSM [68]	Implicit	Stochastic Segment Models	fsync
Digalakis [21]	Split & Merge	Dynamic System	fsync
SESM [54]	Boundary Detector	Cepstral Averages	bsync
OGI Alphabet [16]	Viterbi	Knowledge–based Features	N/A
SWISS^2 [this thesis]	Segment Lattice	Formant Features	bsync

fsync = Frame–synchronous Viterbi search
bsync = Boundary–synchronous Viterbi search

2.1.1 Segmentation Methods

The purpose of the segmentation step is to provide the classification stage with segment–boundary hypotheses that guide the extraction of features spanning potentially an entire segment (see Chapter 1 for a discussion of the merits of segment–based features) or larger speech units such as syllables or words. We distinguish between implicit and explicit segmentation of speech. An explicit segmentation algorithm selects a subset of all possible segmentations either in a bottom–up fashion by merging shorter segment hypotheses, or in a top–down refinement algorithm. Traditional frame–based systems, such as those using Hidden Markov Models implicitly segment the speech into phonetic segments as a result of the Viterbi [25] search or the Forward search [72]. However, they cannot be considered

segmental systems since there is no segment–classification step as part of the recognition process (only frame "classification").

A popular approach to segmentation, initially proposed by Bush and Kopec [9], similarly considers all possible segmentations and therefore segments the speech implicitly. The idea was further pursued by Ostendorf and Roukos [68] in their work with Stochastic Segment Modeling. They proposed a joint segmentation and classification algorithm to estimate the parameters of the segment classifier. However, it was acknowledged that the performance was not realistic for practical purposes. Therefore Digalakis proposed a Split–and–Merge algorithm [22] to reduce the computational requirements. An initial segmentation, based on dividing the speech into equally long segments of average segment length, is subsequently refined by hypothesizing splits and merges of current segment hypotheses while searching for the maximum joint segmentation and classification likelihood. The initial implicit segmentation algorithm (as proposed by Ostendorf and Roukos) is thus transformed into a top–down refinement of an explicit segmentation.

The segmentation process, proposed by Leung et al. [54] as part of the Stochastic Explicit–Segment Modeling (SESM) system, is similar in spirit to the implicit segmentation used in the SSM approach. However, in order to reduce the search space, not all possible segmentations are considered. Instead, the same initialization process as in the SUMMIT system (see below) is used to hypothesize boundary candidates. Once $\mathcal{B} = \{b_k\}$, the set of boundary candidates, is determined in this way, the boundaries b_k are classified using a Multi–Layer Perceptron [76] (MLP). The inputs to the MLP are the averaged mean rate responses [80] of the adjacent 8 seed regions (4 on either side of b_k). The output is either a binary (boundary / non–boundary) classification or the label of one of the possible 36 broad category boundaries. The boundary $b_{\beta_j\beta_{j+1}}$ is located between the broad category segments β_j and β_{j+1}. (Hence, the boundary $b_{\beta_j\beta_j}$ represents either a boundary between two segments of the same broad phonetic category or an inserted boundary within a segment of broad category β_j.)

The above is in contrast to bottom–up algorithms, such as the dendrogram or the

[2]**SWISS** = **S**egmental, **W**ord–**I**ndependent **S**peech Recognition **S**ystem

algorithm proposed in this work (Chapter 3). The dendrogram was initially proposed by Glass [28] and successfully implemented by Zue et al. [94] in MIT's SUMMIT system. As its first step, the dendrogram algorithm divides the speech into acoustic segments, called seed regions, using a sensitive edge detector. This algorithm associates each frame to one of its immediate neighbors. Acoustic boundaries are found whenever the direction of the association changes from the past to the future. The distance metric D^* used for this association is the product of three components to avoid problems of local extrema [93]:

$$D_t^* = \hat{D}(d_{i-1}, d_i)\hat{D}(d_i, d_{i+1})\hat{D}(a_i, a_{i+1})$$

where

$$\begin{aligned} a_i &= \text{acoustic vector of frame i (and later region i)} \\ d_i &= a_i - a_{i+1} \\ \hat{D}(\vec{x}, \vec{y}) &= 1 - \frac{\vec{x}\vec{y}}{|\vec{x}||\vec{y}|} \end{aligned}$$

However, this initial acoustic segmentation tends to find too many additional boundaries (e.g. between the stop burst and the aspiration in an unvoiced stop). Therefore a hierarchical multi–level representation, the dendrogram, is computed. The association algorithm described above is repeated but instead of the frame association, the mean acoustic vectors of the initial seed regions are compared and regions are merged in the direction of the larger association. This process is iterated until the entire utterance is represented by a single region. Finally, an acoustic–phonetic network is created by including all single and paired regions from the dendrogram. The dendrogram algorithm is conceptually similar to the one proposed in this work. However, we use broad category classification scores as the basis of the level–building algorithm, rather than the difference in the mean acoustic vectors.

Chigier and Brennan [13] use knowledge–based rules to reduce the number of hypothesized segments in a dendrogram. They use rules based on region durations, similarity of region spectra, waveform contour, and region classifications into a set of 19 broad phonetic classes to prune the size of the dendrogram. In their work, these rules help reduce the

depth of the dendrogram (total number of segments in the dendrogram divided by the total number of segments in the hand labeled transcription) from 4.5 to 1.8 while slightly increasing the deletion rate from 5.8% to 6.7%.

As an alternative to the above systems that either implicitly or explicitly segment the speech and generate a hierarchical structure of segment hypotheses, one can use an existing frame–based speech recognizer to define one or more segmentations. The OGI Alphabet Recognizer [16, 79] uses a 2–pass approach to recognize the letters of the alphabet in the respective languages (English and French systems, German and Spanish prototypes exist). In the first pass, an MLP classifier, using selected features around the frame to be classified, assigns a phonetic probability score to each speech frame. A Viterbi search is then used to find the most likely segmentation of the speech into letters and phones given pronunciation and durational constraints. The letter pronunciations are expressed in terms of either phonemes or a combination thereof (e.g., the nasal phonemes /m/ and /n/ are merged into a single category /mn/). Certain acoustically difficult distinctions need not be made in the initial segmentation pass since the output of this processing step is a likely segmentation and not the unique identification of the letter. The English system uses 22 such units, the French version 23. The idea of locating and segmenting the letters into the constituents was pushed even further in the German and Spanish prototype versions where only broad phonetic categories were used (e.g., vowel, semi–vowel, nasal, fricative, voiced stop, aspirated stop, closure / silence). Subsequent work has shown that relying on a single segmentation, while feasible for a reasonably constrained task such as alphabet recognition, does not scale well to more complex tasks such as large vocabulary recognition. Austin et al. [3] use a conventional HMM in a first pass of their BYBLOS system to generate N segmentation hypotheses from the N highest scoring recognition paths. The parameter N is chosen to assure that the correct segmentation is included in the list most of the time ($N = 20$).

At this point in time, no single segmentation method has emerged as the solution to all problems. There seems to be a trend towards considering a large number of segment hypotheses (bounded primarily by duration constraints) and letting the segment classifier determine the correct ones via appropriate probability estimates for false positives.

However, most systems do not explicitly train their classifiers on recognition errors (we will outline a method for training on errors in Section 4.4). All systems use a single distance function despite the fact that the acoustic nature of segment boundaries (see the introduction to Chapter 3 for a discussion) are diverse and sometimes ambiguous (e.g. the exact location of a semi–vowel / vowel boundary is largely arbitrary). We outline a context–dependent distance function addressing this problem in Appendix A.

We contend that the role of the segmentation stage should not be restricted to providing segment hypotheses, but rather should extract additional information such as broad phonetic information or measurements that will help in defining a probability for a particular segmentation. This idea is taken up again in Section 7.1.

2.1.2 Segment Classification

The main advantage of using a segment–based recognition framework is the added flexibility and power of whole–segment classification (as compared to the frame–based paradigm). The knowledge of the location of segment boundaries allows for the use of powerful intra– and supra–segmental features. It is a well–documented fact (see, e.g., [21]) that there is a high degree of correlation among parameters of speech frames of a phonetic segment, both in frequency and in time. One of the themes of segment classification research is the modelling of these correlations by using feature representations that can capture the dynamics (trajectories) of the parameters over the duration of the entire phonetic segment.

In addition to the methods described below, there is another line of research focusing on using explicit, knowledge–based features in a rule–based (expert) system framework (see for example [81] and [95]). We believe that a combination of features motivated by speech knowledge and statistical classifiers will ultimately yield the best performance. This approach has been the basis of the research at the Spoken Language Systems Group at the Massachusetts Institute of Technology and resulted in the novel feature extraction mechanism for SUMMIT (see below).

The simplest method of modeling the dynamics of parameter evolution over a segment is to approximate the trajectories by a piecewise–constant function. Typically, the segment is split into thirds (for the same reasons that HMMs use three–state phonetic models to

capture coarticulation effects at the beginning and end of a phonetic segment). The mean values of the parameters over each third are computed and used as classification features (eventually in combination with the segment duration). This approach is used as the baseline method in this work. It was also used by Leung in his thesis work on TIMIT segment classification [51].

Gish and Ng [27] augmented this basic approach by using a quadratic polynomial to model the temporal evolution of the cepstral coefficients $c(n)$:

$$c(n) = b_1 + b_2 \cdot n + b_3 \cdot n^2 + e(n) \qquad \text{for} \quad n = 1, \ldots, N, \qquad e(n) \sim N(0, \Sigma).$$

The classification is then done via the maximum a posteriori probability rule:

$$m^* \;=\; \underset{m}{\operatorname{argmax}} \quad l(k|m, N) \cdot p(N|m) \cdot p(m) \qquad ,$$

where $l(k|m, N)$ is the likelihood of segment k being phoneme m and of length N frames estimated using a Gaussian mixture model, and $p(m)$ is the prior probability of phoneme m.

A state–space dynamical system is used by Digalakis [21] to model the trajectories of acoustic attributes (Mel–frequency cepstral coefficients). The model parameters are estimated using the EM algorithm. The likelihood score estimation for each phoneme is based on the innovation process given by the Kalman filtering.

Similarly, Goldenthal models the trajectories with a non–parametric function [31]. He defines tracks, which describe the temporal evolution of acoustic attributes over a segment, as a sequence of M state vectors T. The track parameters are then used to generate a synthetic segment model $G = f(T, N)$ of length N which is compared against the speech segment S to be classified. The resulting error $E = S - G$ is used to estimate the likelihood of segment S being phoneme α using a Gaussian classifier.

A slightly different approach is taken by Ostendorf and Roukos [68]: instead of modeling the observed variable length trajectories X of length L and dimension k, they first transform X into a fixed–length representation Y of length M using a linear time–warping function T:

$$Y \;=\; X \cdot T_L \qquad .$$

Several methods of resampling are considered: linear time sampling and frequency sampling, with and without interpolation. The model does not make any independence assumptions among the feature vectors (unlike the HMM) for classification. Theoretically, a full covariance matrix of dimension kM x kM could be estimated. However, due to a lack of training data, only block–diagonal covariance matrices for the Gaussian classifiers were computed in the reported experiments. Depending on the arrangement of the features, this means that the block diagonal covariance matrices either model the frequency or the temporal correlations among the features. Their experiments show that estimating the frequency correlations yields superior classification results over the time correlations and complete independence conditions. Using block–diagonal correlation matrices makes this approach equivalent to an HMM with a constrained state sequence.

The uniqueness of the SUMMIT classification mechanism is not the classifier used (Gaussian classifier), but the features: a set of generalized algorithms [70]. These algorithms together with their associated free parameters form a search space. The parameters can be adjusted to optimize the segment classification performance. An example of such an algorithm is the computation of the spectral center of gravity with the lower and upper frequency edges as the free parameters. This measurement tends to follow the second formant, an important feature for the front– versus back–vowel discrimination.

The trend in these classification methods is the modelling of the dynamics of the feature trajectories in an attempt to capture the temporal and frequency correlation among the features. However, the feature space used in these experiments is in the cepstral domain. We believe that a formant or articulatory feature space is better suited to capture the dynamics of speech (see the arguments on spectrogram reading in Chapter 1). Nevertheless, significant improvements from modeling the dynamics have been demonstrated by the above systems.

2.1.3 Search Algorithms

The main design issue facing system designers of segmental systems is the problem that not all the paths through the search space of segment hypotheses contain the same number of segments. Hence the products of path probabilities of competing search hypotheses consist

of different numbers of terms. This would not be a problem if true segment probabilities could be estimated. However in practice, special measures have to be taken to avoid that solutions with fewer segments will have a higher probability (because it is the product of fewer path probabilities). Generally, a global insertion penalty is used to compensate for this effect.

Phonetic recognition finds the most likely sequence of phonetic units A^* by maximizing the posterior probability of the sequence of phones A = $\{a_1, a_2, ..., a_N\}$ given a sequence of acoustic observations X = $\{x_1, x_2, ..., x_T\}$. In segment–based systems, the search is constrained by the segment hypotheses $S_i \in \mathcal{S}$ as follows :

$$A^* = \underset{A}{\operatorname{argmax}}\, p(A|X) = \underset{A}{\operatorname{argmax}} \sum_{S_i \in \mathcal{S}} p(A, S_i|X) \tag{2.1}$$

In practice, the summation in Equation 2.1 is often replaced by the Viterbi algorithm [25] which finds the most likely joint segmentation and classification path:

$$A^* = \underset{A,S}{\operatorname{argmax}}\, p(A, S|X) \tag{2.2}$$

The basic equation 2.2 can be further decomposed depending on the type of classifier used in the segment classification step.

In their SESM system, Leung et al. [54] use neural networks to classify the segment hypotheses. Because standard neural networks (e.g., MLP) estimate posterior probabilities [6], equation 2.2 gets rewritten as:

$$A^* = \underset{A,S}{\operatorname{argmax}} \underbrace{p(A|S, X)}_{\substack{phonetic \\ classification}} \cdot \underbrace{p(S|X)}_{segmentation} \tag{2.3}$$

Let $\{b_1, b_2, ..., b_K\}$ be the set of boundaries, generated by the segmentation mechanism, within the hypothesized segment s_i. Then the probability of the segment s_i is the product of probabilities $p(\overline{b_k}|X)$ that those internal boundaries are not true boundaries. Hence

$$p(S|X) = \prod_{i}^{N} p(s_i|X) = \prod_{i}^{N} \prod_{k \in s_i} p(\overline{b_k}|X)$$

In [55], Leung et al. extended this basic approach to context–dependent phone models and broad category boundary classifications. It should be noted at this point that in

equation 2.3 posterior probabilities are multiplied, which does not correspond to the usual independence assumption.

In his Ph.D. work [31], Goldenthal uses Gaussian segment classifiers, which estimate likelihoods. Therefore equation 2.2 is transformed into:

$$A^* = \underset{A,S}{\operatorname{argmax}} \underbrace{p(X|A,S)}_{\substack{likelihood \\ estimation}} \cdot \underbrace{p(S|A)}_{segmentation} \cdot \underbrace{p(A)}_{\substack{language \\ model}} \tag{2.4}$$

The term $p(S|A)$ in approximated by $p(S)$, which in his work is a function of the number of segments in S ("segment transition weight"). The above basic equation 2.4 is further extended to include transition components of the acoustic score.

Similarly, Ostendorf and Roukos [68] decompose equation 2.2 into the following terms, where Y is the fixed–length model of X:

$$A^* = \underset{A}{\operatorname{argmax}} \underbrace{p(Y|A)}_{\substack{likelihood \\ estimation}} \cdot \underbrace{p(A)}_{\substack{language \\ model}} \cdot \underbrace{L}_{\substack{duration \\ model}} \cdot \underbrace{C}_{\substack{insertion \\ penalty}} \tag{2.5}$$

The reader should note that in this case the segmentation is implicit. Therefore the absence of segment and segmentation probabilities is compensated for by introducing two additional terms: L and C. The duration model L simulates the segment probability and the insertion penalty term C can be interpreted as an approximation to the probability of a segmentation $p(S)$.

Systems that use a frame–based recognizer to generate segment hypotheses in an N–best list typically recombine the original HMM score with the new classification scores using a linear function, the parameters of which are experimentally optimized.

The above overview suggests the following: firstly, the combination of the segmentation and the segment classification scores depends on the type of classifier used (posterior–probability or likelihood estimator). Secondly, the estimation of $p(S)$, $p(S|A)$, or $p(S|X)$ is poorly understood at this point; therefore these terms are generally approximated by global constants (except in the case of SESM).

MIT's SUMMIT system employs yet a different solution to the problem of estimating segmentation probabilities. The transition probabilities in the pronunciation dictionary

(which in the context of phonetic recognition acts similarly to the language model) are estimated in an iterative process. The probabilities are centered around a neutral value of 1.0 and are adjusted depending on whether or not the model was chosen correctly or incorrectly in a comparison of forced versus unforced recognition.

In this section, we have described the probabilistic framework for several systems. The rightmost column of Table 2.1 lists the type of search algorithms (programming model) used to implement the respective probabilistic formulae. The SUMMIT system uses a frame–synchronous Viterbi search in the first pass of a two pass search. The second–pass A* search uses the estimates from the initial Viterbi search to estimate the remaining costs needed to run the A* search efficiently [35]. The SESM system and the system described in this thesis (SWISS) advance the search not frame–synchronously, but rather from boundary hypothesis to boundary hypothesis, hence a boundary–synchronous Viterbi search. Note that not all the active hypotheses are updated at every search increment.

2.2 Formant Tracking Algorithms

In this section we review two formant tracking algorithms in some detail. The first (Section 2.2.1) is the state–of–the–art formant tracker proposed by Talkin [83] in the mid 1980's and implemented as part of the commercial ESPS toolkit [84]. The second algorithm (Section 2.2.2), proposed by Laprie et al. [47], is closely related to the formant tracker described in this thesis.

Early work includes formant trackers proposed by McCandless [61] and Kopec [41, 42]. McCandless' formant tracking algorithm uses the peaks of the linear prediction spectrum [60] as initial formant candidates. In order to have a good seeding for the tracking hypothesis, the algorithm starts in the middle of a voiced segment and works outwards (towards the boundaries) by applying editing functions. A special feature of this algorithm is an enhancement procedure that is designed to recover missing formants by recomputing the spectrum with new parameters.

Kopec uses Hidden Markov Models to track formants. In his experiments, the n HMM states correspond to possible formant values, quantized into n equally spaced bands of

width $4kHz/n$. The emission symbols are taken from an LPC vector codebook. The state transition probabilities may be viewed as an encoding of continuity constraints on formant motion. He describes both single–formant and multi–formant models and concludes from experiments on the a digit database that there is no performance advantage from simultaneously tracking multiple formants using his models.

2.2.1 ESPS Formant Tracker

In his report, Talkin acknowledges the importance of incorporating non–local constraints into the formant tracking algorithm [83]. However, he believes that Kopec's [42] implementation of a global constraint that formants be continuous is not the best solution, since a global constraint tends to be too weak in sonorant regions and too strong across vowel–consonant boundaries. Therefore, in his system, he modulates the transition cost, which determines the importance of the continuity constraint, by an estimate of the signal's stationarity. He further points out that there are three basic methods of generating formant peak candidates: analysis–by–synthesis [66], peak picking in a smoothed spectrum obtained by LPC analysis (with cepstral smoothing), and solving for the root of a linear predictor polynomial. Specifically, he chose to use the complex roots of the denominator polynomial of the z transform of a linear predictor as the source of formant candidates. The all–pole, linear model is a reasonable model for vowel sounds, and it tends to slightly overgenerate candidates which is desirable because it helps in dealing with noise, spectral zeros and other modeling errors.

The cost function to be optimized by the Viterbi search consists of two parts: local mapping costs and transition costs. The local mapping cost is a linear combination of formant bandwidth (narrower peaks are better), a bias cost giving preference to lower frequency formant candidates, and a term measuring the deviation of the proposed formant location from the expected location for that formant. The transition cost is a quadratic cost function of the relative formant frequency change from the last frame scaled by an estimate of the signal's stationarity. This modulation adds flexibility to the continuity constraint and is a major contribution of his work. All the coefficients for the linear function are determined empirically by multiple supervised passes through a digit database.

Talkin points out that most of the errors encountered in testing occurred in short vowel segments in the context of nasals, strong fricatives and other sounds not well represented by a low–order, all–pole model. He further outlines improvements to the stationarity measure and the initial signal processing stage to better cope with female voices and formant merges.

2.2.2 CRIN / INRIA 2–Pass Formant Tracker

The main objective for the formant tracker proposed by Laprie et al. [47] is to find the set of formant locations that explains the most energy. The algorithm was later extended to add a smoothness term to the objective function [49].

The basic algorithm uses five processing steps. First, elementary tracks (formants or parts thereof) are identified in a process similar to the first stage of the formant tracker described in the thesis (Chapter 5). Next, the elementary tracks are labeled in terms of formants using energy and degrees of acoustic constraint satisfaction (constraints on F1–F2 and F2–F3). The formant candidates thus labeled are then connected end to end to form N 3-tuples of formant interpretation hypotheses. The objective of the formant tracking algorithm is to optimize a cost function E that incorporates aspects of closeness to the data (explaining the most energy) and regularity of the formant tracks:

$$E = -E_{Formant} + \lambda E_{Smoothness} \quad .$$

The above functional is minimized by using an algorithm proposed by Kass et al. [38]. To avoid formant merges by the regularization algorithm, peak enhancement algorithms are applied where needed. Finally, a global assessment of the solutions found by the preceding processing steps based on energy, expected domains for F1–F2 and F2–F3, as well as formant levels is performed to find the single best interpretation.

Laprie et al. report encouraging results of their formant tracking algorithm using visual inspection of the formant tracks overlayed on a wideband spectrogram for stop–vowel tokens for 4 male speakers. They also report that the algorithm runs in a few times realtime on a moderately powerful workstation [49].

2.3 Perceptual Experiments and Formant Theory

In this section we briefly sketch some results reported in the literature describing vowel perceptual experiments and theories of (vowel) perception. This body of work suggests some of the perceptually important features which are incorporated into the formant–based vowel classifier described in Chapter 4.

The entire literature on vowel perception is far too extensive for the purpose of this review. Therefore, we restrict ourselves to the articles resulting from an Acoustical Society of America meeting in 1987 that brought together some of the leading researchers in the field. The purpose of the meeting was to contrast the prevelant theories and to outline open questions and future work.

The article by Strange [82] provides a good overview of the major theories of vowel perception. The classic textbook model of vowel perception describes the vowel characteristics in terms of formant targets extracted from the spectral cross section of the steady–state portions of the vowel. These target locations are either fully attained in sustained vowels, or represent target configurations for the articulators in the case of continuous speech. Often the first two formants, F1 and F2 contain the primary and sufficient information for the vowel identity. However, this model of formant targets has two major problems which Strange terms "speaker normalization" and "target undershoot". The formant–ratio theory has been proposed to solve the speaker normalization problem (see Miller [65] for a historical review of formant–ratio theories). More recent work has been focusing on psychophysically motivated transformations of formants and fundamental frequencies in an attempt to create a feature space (mostly in the F1 versus F2 plane) with less overlap among the vowel categories for all ages and genders. Miller extends the formant–ratio theory by making the interpretation of the formant ratios F2/F1 and F3/F2 dependent on the F1/SR ratio, where SR is a function of the average pitch and tries to capture the speaker characteristics. He constructs a three–dimensional auditory–perceptual space and shows that sustained monophtongal vowels of American English can be segregated with 93% accuracy. (Note: For our own classification experiments, we will use the average pitch over the vowel segment as one of the classification features.) Lindblom and

Studdert–Kennedy [57] addressed the target undershoot problem by demonstrating that perceptual boundaries shifted as a function of the syllable duration (in a CVC context) and the direction of the F2 transition. In related research in the early 1960's, Peterson and Lehiste [69] report that vowels differ in their intrinsic durations. They also noticed systematic differences in the relative durations of onglides (formant transitions into the syllable nucleus), offglides, and quasi–steady–state portions of CVC syllables for tense and lax vowels. Tense vowels tend to have on– and offglides of similar, short duration, whereas the lax vowels tend to have relatively long offglides. (Note: We will define F1 interval features in our classification experiments in Section 6.1.)

A series of perceptual experiments using the TIMIT database were conducted by Cole et al. [15]. They report 54.8% correct vowel identification for the 16 TIMIT vowels when played to subjects in isolation. The identification rate improved to 65.9% when acoustic context was provided to the listener. A related experiment testing speaker normalization effects showed a small but significant increase in listener–labeler agreement. This study did not test consonants. In a similar series of experiments reported by Cole et al. [19], phonetically balanced TIMIT sentences were altered by replacing the acoustics of either the vowel or consonant segments with white noise. Subjects were then asked to identify the words contained in the utterance. In this experiment they found vowels to be more important than consonants for word recognition.

2.4 Segment Classification Experiments

We next review the results reported in the literature for segment classification (vowels and phonemes) as well as phonetic recognition. It should be noted that even though all of the results reported below were established using the TIMIT database, the training and test sets along with the set of recognized phonemes might differ from researcher to researcher. Consult [31] for a good overview of these differences.

Most of the early segment classification experiments focused on vowel classification. This is generally considered to be the more difficult task compared to consonant classification. Table 2.2 summarizes the vowel classification results obtained by several researchers.

Table 2.2: Overview: Vowel Classification Results

Segment Classification Results : Vowels

Researcher	Feature Set	Accuracy
Meng	Auditory Model	64.5%
Carlson & Glass	Formants from Analysis–by–Synthesis	62.6%
	+ gender information	65.6%
Goldenthal	Tracks of cepstral features	66.6%
	+ gender–specific models	68.9%

Using the outputs of an auditory model, Meng [63] achieved 64.5% on a 13–vowel classification task and 65.5% on a 16–vowel task [62]. Carlson and Glass [11] report a 62.5% classification accuracy using averaged Bark spectral vectors and an MLP classifier. They also describe a formant representation based on an analysis–by–synthesis procedure. Averaging the formant locations over segment thirds (3x3 = 9 features) they achieve 56.6% accuracy. Adding formant amplitude and measures of the formant transition speed, the performance was improved to 62.6%. This result will serve as a point of reference for our vowel classification experiments using explicit formant features (see Chapter 4). They report further improvements to 65.6% when providing the MLP classifier with (explicit) gender information. Even better results are reported by Goldenthal [31]. Using the track representation and a Gaussian classifier, he achieves 66.6% correct classification, and 68.9% when using gender–specific models, where the gender in unknown during testing.

Vowels, via the formant structure, carry the most speaker–dependent characteristics which tends to make the classification more difficult. Removing them via speaker normalization techniques or the incorporation of gender information (e.g., gender–specific models) can therefore lead to improved classification accuracy as shown above.

The phonetic segment–classification results for context–independent models are summarized in Table 2.3. Again, it should be noted that the training and test conditions are different among the systems and therefore the results should not be used for direct

comparison.

Table 2.3: Overview: Phonetic Segment Classification Results

Segment Classification Results : Phonemes

Researcher	Feature Set	Accuracy
Chigier et al.	MLP & PLP	78.0%
Goldenthal	Tracks	76.8%
Digalakis	Dynamic System Models	73.9%
Leung	Acoustic Attributes	76.0%

Leung in his Ph.D. thesis [51] experimented with k–Nearest–Neighbor and Gaussian classifiers as well as neural networks in the context of phonetic segment classification. He reports achieving 72% correct phoneme classification on TIMIT for 38 phonemes [53] using the synchrony envelopes and mean–rate responses of Seneff's auditory model [80] along with the segment duration. The best result of 76% is achieved by using 82 acoustic attributes as features to a MLP. Chigier et al. [14] and Leung et al. [52] also experimented with various signal representations and classifier technologies. Their best result of 78.0% was obtained using PLP features [33] and a neural network classifier. Furthermore, on male–only training and test sets, Digalakis [23] achieved 73.9% using his Dynamic System Models.

2.5 Phonetic Recognition Experiments

Lee initially used the TIMIT [46] database for his phonetic recognition experiments using SPHINX [50]. Since then, many researchers have used the same task to compare the performance of their own systems. See Goldenthal's Ph.D. thesis [31] for a good overview of the best results achieved. He points out the differences in training and test sets used by various sites. We summarize the results in Table 2.4 without repeating the differences here.

Table 2.4: Overview: Phonetic Recognition Results using the TIMIT database.

Phonetic Recognition Results

System	Accuracy (39 Phones)	Accuracy (47 Phones)
HMM – Lamel & Gauvin	69.1%	
SUMMIT – Phillips & Glass	68.5%	
Anti–Phones – Glass et al.	69.5%	
STM – Goldenthal	69.5%	
REPN – Robinson	73.9%	64.7%
CVDHMM – Ljolje		69.4%

Lately, Ljolje [59] has reported a recognition accuracy of 69.4% on a 49 phone set and a slightly different scoring of silence and closure segments. He points out that the REPN [75] system achieves 64.7% if the same scoring mechanism would be used. His system, a continuous variable duration HMM (CVDHMM) with context clustering of quasi–triphonic model states, uses a trigram language model to further improve on the recognition performance. Additionally, Glass et al. [30] achieved 69.5% recognition accuracy using anti–phones in a segment–based recognition framework. Note that the results reported in Table 2.4 are established using context–dependent phoneme models in combination with either bigram or trigram language models.

Chapter 3

Segmentation

In the introduction to this thesis, we argued for the use of a segment–based speech recognition system because of its superior segment modeling capabilities. However, the drawback of this approach is a loss of flexibility: once the speech is broken up into segments, it is very difficult for the phonetic search to recover from segmentation errors, especially segment deletions. Therefore, the quality of the segmentation algorithm is crucial to the overall success of the recognition system.

The purpose of the segmentation process is to establish acoustic landmarks, hypothesize possible segmentations of the incoming speech into phonetic segments, and to provide the segment classifier with possible segment boundary locations that can be used to extract knowledge–based features (see Chapter 6 for a discussion of the merits of this approach).

Spectrogram reading experiments have shown [17, 20, 92] that an expert spectrogram reader is capable of locating essentially all segments (97% in continuous speech and 100% for isolated words) found by phoneticians who had access to the acoustics along with the spectrogram. Unfortunately, attempts at mimicking the behaviour of the expert with a computer algorithm have failed, partially due to intrinsic difficulties of time scaling. Certain (phonetic) boundaries can be characterized by a large change in the spectral shape, mostly prompted by a change in the manner of articulation (silence, frication, voicing, nasalization). However, there are also boundaries that only change in the place of articulation (e.g. velar, alveolar, or palatal), by "slowly" moving the articulators. This process results in a definite, but "slow" (approximately 30 – 50 ms in comparison to the common frame rates of 5 or 10 ms), change in the spectral characteristic. A typical example is the semi-vowel/vowel boundary shown in Figure 3.1; one can argue that the

transition from the retroflex /r/ into the front vowel /iy/ starts at time 140ms and ends at 180ms. In contrast, the /s/–/iy/ boundary falls in the time range of 360ms to 370ms.

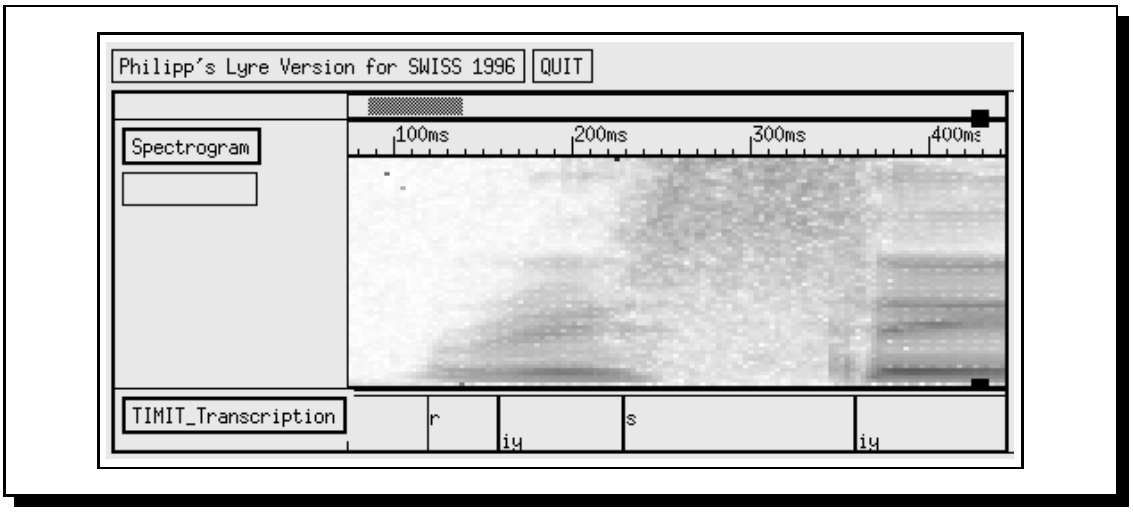

Figure 3.1: Example of "slow" (/r/–/iy/) and "fast" (/s/–/iy/) boundaries

A popular approach to segmentation, initially proposed by Bush and Kopec [9], is to consider all possible segmentations. While this is theoretically the optimal solution, it has been acknowledged that the computational requirements make an implementation thereof impractical at this time [22]. Therefore, a subset of the potential boundary locations is generally selected for further processing. In this work, such a segmentation is called an *Acoustic Segmentation (AS)*. In the context of the dendrogram algorithm [28], these segments are called seed regions.

Most of the segmentation methods rely on some sort of distance measure to detect spectral changes which would indicate the presence of a phonetic boundary between two adjacent frames (see Appendix A for a discussion of commonly used distance metrics). The boundaries between adjacent acoustic segments are then proposed at local maxima of this distance function. As can be seen from the examples above, distinguishing between true boundaries and false positives using a distance function is a difficult problem. Often, the spectral change (from frame to frame) within a segment can be larger than across segment boundaries. In practice, this problem is generally solved by setting a low detection threshold and accepting an oversegmentation of the utterance, followed by a grouping or

level building stage. MIT's SUMMIT system [94] uses the average spectral distance to merge adjacent regions into a hierarchical structure called a dendrogram [28] (see Section 2.1.1 for a summary).

The segmentation algorithm developed in this research is similar to the dendrogram algorithm . However, instead of using distances between mean acoustic vectors to decide on merging adjacent segment hypotheses, we classify the initial acoustic segments into 6 broad phonetic categories and use a set of rules based on the classification scores to make merging decisions. The resulting segment lattice contains segment hypotheses with associated broad phonetic identities. This fact can be used for classifier selection (e.g., separate vowel and consonant classifiers), feature extraction (e.g., provide broad phonetic estimates for quasi–triphone models), and segment lattice enhancement (e.g., insertion of deleted voiced stop segments by rule), as described in Section 3.4.

In the next section, we describe the segmentation of an utterance into acoustic segments. The rules used in the construction algorithm are based on broad phonetic classification results, which are described in Section 3.3. Finally, the construction of the lattice containing the segment hypotheses is described in Section 3.4 and evaluated in Section 3.5.

3.1 Acoustic Segmentation

Most of the segmentation methods rely on some sort of distance measure $d(\vec{x}, \vec{y})$ to detect spectral changes. The most common metric is the Squared Difference Function:

$$d(\vec{x}, \vec{y}) = \quad (\vec{x} - \vec{y})^T A(\vec{x} - \vec{y}) \qquad \text{Squared Difference Function}$$

where $\vec{x}$ and $\vec{y}$ are feature vectors representing the spectrum to the left and the right of the potential boundary respectively. The matrix A defines the distance metric used. Popular choices are $A = I$ for the Euclidian distance and $A = Cov^{-1}$ (the inverse of the feature covariance matrix) for the Mahalanobis distance [26].

In the past, we have experimented with other distance metrics, such as the Spectral Variation Function and a context–dependent extension thereof. Appendix A contains

an overview and discussion of those alternatives. Initial experiments with the context-dependent spectral variation function were encouraging, but failed to yield any substantial performance improvements. Therefore, we decided to use the squared difference function with the Euclidian metric as the basis of the segmentation algorithm in this work:

$$d_0(t) = d(t, X_0^T) = \left(X_{t-L}^t - X_t^{t+L}\right)^T \left(X_{t-L}^t - X_t^{t+L}\right)$$

where X_0^T is the vector of acoustic feature vectors (Mel–Frequency Cepstral Coefficients in this work) from frame 0 up to but excluding frame T and X_{t-L}^t the subset of feature vectors starting at frame $(t - L)$ and ending before frame t.

It should be noted that the choice of a suitable distance function has to be made in the context of the segmentation algorithm. The lattice generation algorithm used in this research for example is able to insert missing /stop/ segments and omitted /vowel/–/vowel/ boundaries by rule and hence doesn't require the distance function to reliably extract those boundaries (both of which are difficult cases, either due to the short duration or the small frame–to–frame changes).

The acoustic segmentation is subsequently constructed by a peak picking algorithm on the distance function $d_0(t)$ with thresholding on the height of the peak. It should be noted that one of the consequences of this choice (finding maxima) is that phonetic segments of length one (frame) will not be identified by this algorithm, something which occasionally occurs for short voiced stops such as /b/ or /d/ in front of vowels. As will be explained in the next section, short stop segments will be inserted by rule into the lattice wherever there is a /closure/–/vowel/ transition without an intervening /stop/ segment.

In practice, we have observed a considerable number of false positives (incorrectly proposed segment boundaries) in silence and fricative segments. Therefore, we have modified the original distance function d_0 to include square difference terms of energy (E) and zero crossing (ZC) information:

$$d_1 = f\left(d_0, E_0^T, ZC_0^T\right) \qquad .$$

3.2 Evaluation of Boundary Detection

Normally, the efficiency of a boundary detection algorithm is evaluated by observing how many times there was a machine generated boundary within e.g. 10 ms of the hand labeled phonetic segment boundary. Since we are interested in assessing the usefulness of a particular boundary detection algorithm as a basis for a segment building algorithm, we are interested in the number of useful segments that we will be able to construct based on the initial segmentation. Therefore, we chose to adopt a different scoring scheme, wherein each machine generated boundary is associated with the nearest hand–labeled phonetic segment boundary. Thus, all segmentation errors are counted as deletions, rather than insertions errors. Figure 3.2 shows an example of such an assignment.

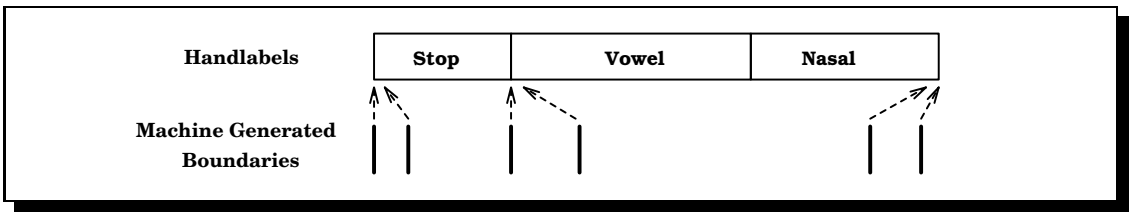

Figure 3.2: Illustration of the Boundary Detection Scoring Process

In this case (Figure 3.2), the phonetic boundary between the vowel and the nasal segment doesn't have a corresponding machine generated boundary hypothesis and would thus be counted as a missed boundary. The same figure also shows that there might be more than one hypothesized boundary per true boundary. By assigning the machine generated boundaries to true boundaries and then counting the number of boundaries that were left uncovered, we can estimate the degree to which we will be able to build a segment lattice that has the same structure as the true (hand labeled) segmentation.

We have observed that most of the boundary deletions happen when "slow" phonetic boundaries remained undetected and a single long acoustic segment was constructed instead. Adding a post–processing rule that splits each acoustic segment longer then a global threshold *ASSplitDur* resulted in a markable improvement as can be seen in Table 3.1, which summarizes the boundary detection results.

The overgeneration (number of proposed acoustic boundaries per true hand–labeled

Table 3.1: Acoustic Segment Boundary Detection Results (overgeneration is the number of acoustic boundaries per TIMIT boundary).

Boundary Detection Results

Distance Function	Overgeneration	Missed Boundaries
d_1	1.82	7.0%
+ $ASSplitDur = 3$ frames	3.27	2.8%
+ $ASSplitDur = 5$ frames	2.32	3.9%

phonetic boundary) is inverse proportionally to the deletion rate: the higher the overgeneration, the lower the deletion rate will be. For our further experiments in this thesis, we set $ASSplitDur = 5$.

3.3 Acoustic Segment Classification

The neural network used to classify acoustic segments computes the probability b_i that the acoustic segment as_i is part of a phonetic segment with broad category identity β_k:

$$b_i = P(\beta_k | as_i, X) \qquad .$$

The broad categories are based on the manner of articulation categorization:

$$\beta_k \in \{\text{closure, vowel, semi–vowel, nasal, fricative, stop}\} \qquad .$$

Figure 3.3 summarizes the classification features for the neural network classifier resulting in a 62–dimensional feature vector.

The classifier was trained on every 4th utterance and evaluated on every 8th utterance of the training part of the TIMIT database. The correct label for each acoustic segment was inferred from the phonetic label of the TIMIT segment overlapping it the most. The overall classification accuracy was 84.0% with individual performances varying from 45% for semi–vowels to 93% for vowels.

Acoustic Segment Classification Features

- average MFCC over each half of as_i (14+14 features)
- average MFCC over the left and the right neighboring acoustic segments as_{i-1} and as_{i+1} (14 + 14 features)
- average energy and zero crossing over as_{i-1}, as_i and as_{i+1} (6 features)

Figure 3.3: List of the acoustic segment classification features.

It should be noted at this point that the classification accuracy is not crucial to the quality of the segment lattice as long as the classification score (network output activation) of the correct category is above a certain threshold (see Section 3.4 for more details).

3.4 Construction of the Segment Lattice

As pointed out earlier, the acoustic segmentation algorithm might break a phonetic segment into multiple acoustic segments. Hence the purpose of the lattice construction algorithm is to reassemble the acoustic segments into phonetic segments [1].

We realize that it is virtually impossible to generate an unique true segmentation. This processing step therefore constructs a lattice of segment hypotheses containing alternate segmentations in regions where the acoustic signal is ambiguous. The segment classifier is expected to produce low phoneme classification scores for those segments that either span multiple true phonetic segments or only parts of a segment, which would result in them not being considered favorably by the phonetic search.

Because of the nature of the search algorithms used in this research (modified Viterbi search, see Chapter 7), we will have to ensure that the lattice–building algorithm generates the start of at least one segment hypothesis whenever the end of a segment is hypothesized. This is required because the search algorithm will have to account for exactly one phonetic

[1] There is the possibility of using the acoustic segments directly for recognition similar as in [91]. In this work however, we will use phonemes as the sub–word units for recognition.

identity for each speech frame (no overlap of phonetic segments and no frames without a covering phonetic segment).

The basic idea of the lattice generation algorithm is to maintain a stack of active hypotheses about the start time and the broad phonetic identity of possible segments. The algorithm considers one acoustic segment at a time and decides for each of the hypotheses on the stack if the new acoustic segment further supports the hypothesis or if it should be terminated. This decision is based on the classification scores of the broad–category segment classifier.

The lattice–building algorithm uses a simple thresholding mechanism as a decision criterion. Therefore, the acoustic segment classification accuracy is not as important, as long as the classification score of the correct category is above the global threshold θ.

Let $\mathcal{H} = \{h_k\}$ be the set of active hypotheses h_k at any given time. The lattice–building algorithm is summarized in terms of $\mathcal{H}$ in Table 3.2.

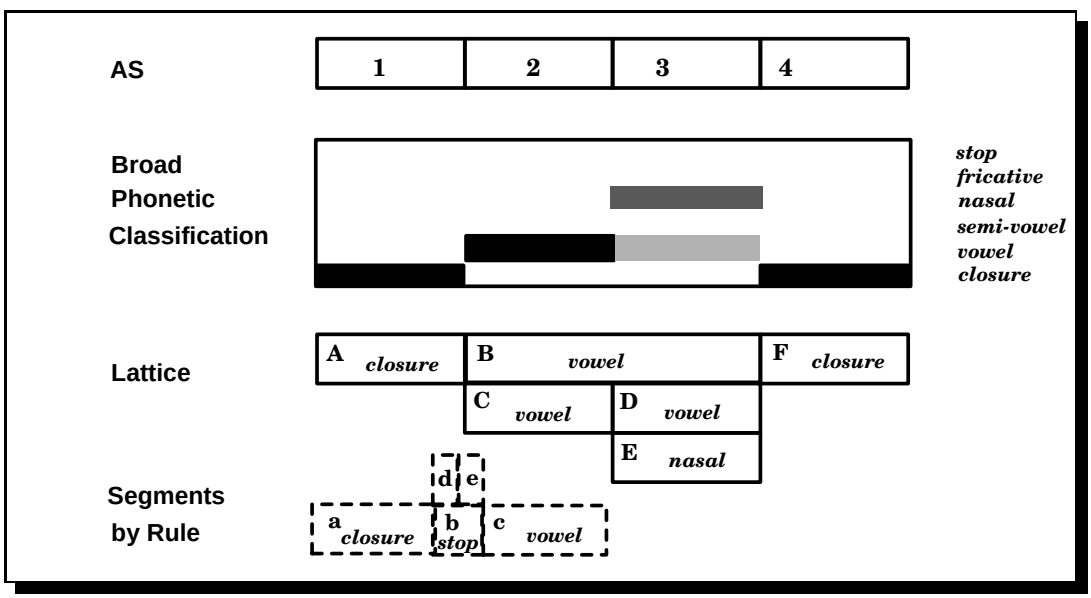

Figure 3.4: Illustration of the Lattice Generation Rules: (from top to bottom) Acoustic Segmentation, Broad Phonetic Classification of AS (the darkness of the bar representing b_i, the higher the output activation of the AS classifier for b_i), Segment Lattice with segment hypotheses A–F, and additional segment hypotheses a–e inserted by applying the stop–insertion rule.

Table 3.2: Lattice Generation Rules

Lattice Generation Rules

1. Initialize the set of hypotheses $\mathcal{H}$ by generating hypothesis h_0 as starting at time 0 and having the broad category identity equal to the index of the highest output within the acoustic segment a_0.
2. Advance to the next acoustic segment as_n.
3. For each hypothesis h_k currently on the stack determine if the classification score of the broad category of h_k in as_n is below the threshold θ. Then terminate h_k, otherwise continue the hypothesis.
4. For each hypothesis h_k currently on the stack terminate it if the maximal duration $SplitDur$ is exceeded (currently there is just one global maximal duration of 5 frames).
5. For each broad category β_j not considered in step 3 check if the classification score in as_n is above the threshold θ. If true then add a new hypothesis onto the stack with start time b_n and identity β_j.
6. Make sure that if a hypothesis was terminated then at least one new hypothesis was generated. If that is not the case then start a new hypothesis, using the highest scoring broad category.
7. Make sure that if a new hypothesis was generated then at least one active hypothesis was terminated. If that is not the case then hypothesize the termination of all hypotheses currently on the stack as well as keeping them on the stack.
8. Repeat step 2 until the end of the utterance is reached.
9. Terminate all remaining active hypotheses.

Figure 3.4 illustrates the operation of the lattice generation algorithm. Initially, $h_1 = \{0, closure\}$ is hypothesized because for the acoustic segment as_1 the /closure/ category has the highest probability. No other hypothesis is generated at this point since no other category's activation is above the threshold (Rule 1). Next, the algorithm is advanced to as_2 (Rule 2), where we check whether or not the hypothesis h_0 still holds, that is whether /closure/ is still a viable hypothesis. Since this is not the case, h_1 is terminated and segment A is inserted into the segment lattice (Rule 3). However, the classification scores indicate that as_2 belongs to /vowel/ and hence we hypothesize $h_2 = \{1,\text{vowel}\}$ (Rule 5). The conditions in Rules 6 and 7 are satisfied and so the search is advanced to as_3. Similarly, segment hypotheses D, E, and F are created and added to the segment lattice. After extensive browsing of the training data, the activation threshold θ was set to 0.1 and the maximal duration $SplitDur$ to 5 frames.

We have observed that many of the short, voiced stops (/b/ and /d/) are omitted by this segmentation algorithm. Since these deletions occur mostly in a /closure/–/vowel/ context, they can easily be reinserted by rule at this point. Whenever a /closure/ segment is terminated and no /stop/ segment is started at the same time, a possible /stop/ segment is inserted by rule. This is illustrated again in Figure 3.4. The segment hypotheses a–e are added to the lattice as a result of the application of the stop–insertion rule.

Additionally, we augment the lattice using the following two mechanisms: splitting long segment hypotheses (to anticipate missed "slow" boundaries), and merging adjacent segments if the lattice only contains a single, short segment (to reduce insertion errors). See Section 7.2.2 for an example.

3.5 Evaluation

In this section, we assess the quality of the segment lattices generated by the algorithm described above. The assessment has to consider two competing goals of a segmentation algorithm: on one hand, each true (hand–labeled) segment should be maximally overlapped by a machine–generated segment hypothesis, but on the other hand, the total number of hypothesized segments should be small to minimize the number of false positives in the

phonetic search.

As with the formant tracker described in Chapter 5, the true measure of the quality of a segment structure, such as the segment lattice described in this chapter, is reflected by the success of the phonetic recognition. As an approximation, Chigier and Brennan [13] use statistics of insertions, deletions, correct segments, and depth to assess the performance of segmentation algorithms. Depth is defined as the total number of segments hypothesized by the algorithm divided by the total number of segments in the hand–labeled transcriptions. They compute insertions and deletion statistics of the best path through the network, where the best path is defined as the path that minimizes the boundary differences to the hand–labeled transcription as well as the insertions and deletions of segments. Table 3.3 summarizes the statistics for the lattice generation algorithm described in this chapter. The results were computed on 460 utterances from the training set not used to train the acoustic segment classifier. There were a total of 18327 TIMIT segments contained in this evaluation set.

Table 3.3: Lattice segment alignment statistics. The dendrogram numbers were computed by Chigier and Brennan.

Alignment Statistics

System	Insertions	Deletions	Correct	Depth
Baseline	2.48%	1.60%	98.3%	6.0
+ Stop Insertions	2.53%	0.98%	98.9%	6.6
+ Augmentation	1.22%	1.08%	98.9%	7.0
Chigier & Brennan [13]	11.00%	6.60%	93.3%	1.8
Dendrogram	2.20%	5.80%	94.2%	4.5

The performance numbers by Chigier and Brennan for their rule set and the original dendrogram (computed on a different test set) are provided as a reference. Note that they chose different operating points (low depth) than this work, which is aimed at reducing deletion errors while achieving a high degree of overlap with the true labels.

Additionally, we also performed a straightforward alignment of the hand–labeled transcription with the segment lattice. If we require 70% overlap between a hand–labeled segment and a machine–hypothesized segment then the analysis shows that 93% of all TIMIT segments do have a corresponding overlapping machine–generated segment.

3.6 Summary

In this work, we propose to use an explicit–segmentation algorithm. We use a distance function based on the Euclidian Squared Difference metric to segment the utterance into acoustic segments which are subsequently classified into 6 broad phonetic categories. The lattice construction algorithm uses rules to merge acoustic segments into phonetic segments and to insert additional segment hypotheses as needed (stop insertions). This algorithm produces a lattice containing segment hypotheses with a broad phonetic identity that substantially overlap (by more than 70%) 93% of the TIMIT hand–labeled phonetic segments. Applying a stop–insertion process and augmenting the lattice with additional segments in places of likely segmentation errors, we can generate segment lattices with reasonable insertion and deletion rates while keeping the total number of segment hypotheses manageable.

Chapter 4

Segment Classification

In this thesis, we report on phoneme classification experiments using the TIMIT [46] database. A series of classifiers with increasing complexity, from a baseline system using static features to separate vowel and consonant classifiers using explicit formant features, are described.

After explaining the training and test sets used for these experiments in Section 4.1, we describe the baseline classifier: a phoneme classifier using mel–frequency cepstral coefficients (MFCC) extracted and averaged over selected regions of the segment, along with average energy and zero crossing rate. Next, separate classifiers for sonorants (vowel and semi–vowels) and consonants using averages of MFCC, energy and zero crossing rate are evaluated. In Chapter 6 we describe classification experiments using explicit formant features.

The classifier architecture used in this research is the Multilayer Perceptron (MLP), trained with a conjugent–gradient optimization algorithm ([4, 5]). We believe that the discriminant nature of this classifier is a major advantage over other classification methods. In his Ph.D. thesis, Leung [51] compared various classifier technologies on a phonetic segment classification task and found MLPs to perform the best.

In our work, we have noticed a discrepancy between the training and testing conditions for these classifiers. Therefore, in Section 4.3, we report on experiments using machine–generated segmentations in training and testing of the baseline classifiers. A few thoughts on training on errors in Section 4.4 round off this chapter.

4.1 Data Sets and Signal Processing

For our classification experiments, two different training sets are used: in addition to the standard NIST training set (NIST–Train), we have further divided NIST–Train into a training (OGI–Train) and two development test sets by selecting every second and every fourth speaker, respectively from NIST–Train. This guarantees that the sets contain similar speaker distributions from all 8 dialect regions as well as a balanced gender distribution. Table 4.1 summarizes the statistics of the training and test sets used in our classification and recognition experiments.

Table 4.1: Training and Test Set Statistics

Training and Test Set Statistics

Set Name	Speakers			Utterances
	M	F	T	
OGI Training Set (OGI–Train)	161	70	231	1848
OGI Development Set 1 (OGI–Dev1)	84	32	116	928
OGI Development Set 2 (OGI–Dev2)	81	34	115	920
MIT Dev Test (MIT–Dev)	34	16	50	400
NIST Training Set (NIST–Train)	326	136	462	3696
NIST Final Test Set (NIST–Core)	16	8	24	192

The TIMIT labels were reduced to a set of 40 categories (Table 4.2), the 39 phonetic categories suggested by Lee [50] and a separate voiced closure class. This division into voiced and voiceless closures was made to reduce the within–category variance. Additionally, the nature of the preceding closure might be used as a feature in a dedicated stop classifier.

In this research, unless stated otherwise, we represent the signal with the mel–frequency cepstral coefficients of order 14 with a window size of 25.6 ms and an increment (frame size) of 10 ms. The preemphasis factor is 0.97.

Table 4.2: Phoneme Set

Phoneme Sets

Vowels and Semi–Vowels										
iy	ih	eh	ae	ah	uw	uh	aa	ey	ay	oy
aw	ow	er	l	y	w	r				
Consonants										
m	n	ng	ch	jh	dh	dx	b	d	g	p
t	k	z	v	f	th	s	sh	hh	cl	vcl

4.2 Baseline Classification Experiment

The purpose of the baseline experiments is to establish initial classification performance numbers. The classification features are similar to the ones proposed by Leung [51]. They are static in nature: the average MFCC, energy and zero crossings of the segment thirds and selected frames to the left and right of the current segment (Figure 4.1). The dimension of the input feature vector for the neural network classifier is 113.

Baseline Features

1. Average MFC, Energy, and Zero Crossing over segment thirds $[(14+1+1)*3 = 48$ features$]$,
2. Average MFC, Energy, and Zero Crossing for the two frames immediately to the left, respectively right of the segment $[(14+1+1)*2 = 32$ features$]$,
3. Average MFC, Energy, and Zero Crossing for the three frames adjacent (left and right) to the frames in 2 $[(14+1+1)*2 = 32$ features$]$,
4. Segment duration [1 feature]

Figure 4.1: Features for the Baseline Segment Classifier

Neural network classifiers (MLP) with varying hidden layer sizes were trained and

tested on the hand–labeled segmentations. The results reported are the best performances achieved for all architectures. In addition to phonetic segment classifiers, we also trained separate vowel and consonant classifiers to be used later in recognition experiments involving multiple classifiers. Therefore, we have trained vowel/semi-vowel and consonant classifiers with the same input features as the phonetic classifiers but with 18 outputs (14 vowels and 4 semi–vowels) for the vowel classifier and 22 outputs for the consonant (and closures) classifier. The classification results are summarized in Table 4.3.

Table 4.3: Classification Results for Baseline Classifiers

Baseline Classification Results

Classifier	Training	Testing	Accuracy
Baseline Phoneme	OGI–Train	OGI–Dev1	71.4%
Baseline Phoneme	NIST–Train	MIT–Dev	**74.1%**
Baseline Vowel / Semi–Vowel	OGI–Train	OGI–Dev1	66.0%
Baseline Vowel / Semi–Vowel	NIST–Train	MIT–Dev	**71.6%**
Baseline Consonant	OGI–Train	OGI–Dev1	74.1%
Baseline Consonant	NIST–Train	MIT–Dev	**78.8%**

The classification results for the baseline system are comparable to results reported elsewhere (see Section 2.4 for a overview). It is interesting to note the difference in performance between the classifiers trained in OGI–Train compared to NIST–Train. The effect is mainly due to a difference in the size of the training sets: OGI–Train contains half of the training material of NIST–Train.

4.3 Training and Testing on Machine–Generated Segments

One of the keys to successful classifier design is the matching of training and testing/deployment conditions [85]. To this point, the features for the segment classifiers were extracted with the knowledge of the true locations of the phonetic segment boundaries. However, when

using the classifiers in a phonetic recognition system, we do not have access to the hand–labeled segmentations. Instead we have to use the segmentation mechanism described in Chapter 3, which produces segmentations that can be slightly different from the hand–labeled ones. In order to account for this mismatch between training and usage conditions, we investigate appropriate procedures for the training and testing of our segment classifiers on machine–generated segmentations.

In order to extract training feature vectors based on machine–generated segmentations, we need to select those lattice segments which have the most overlap with the hand–labeled segments and form a legal path through the segment lattice. Doing so should maximize the correct network outputs for machine–generated segment hypotheses similar to the TIMIT segments, and thus increase the likelihood of finding the correct path in the phonetic search. In order to determine the candidates for the machine–generated segments, we first constructed the segment lattice (see Chapter 3 for more details) for each training utterance. Next, a search through the lattice was done with the objective to minimize the distance between the boundary locations of the aligned segmentation with the hand–labeled segmentation. At the same time, this forced–alignment process also provides the labels for the training of the classifiers. Figure 4.2 shows an example of this procedure. It demonstrates the differences between the hand–labeled (TIMIT) and machine–generated (ALIGN) segmentations, most notably for the utterance initial nasal /n/.

The phonetic classification performance for various training and testing conditions is summarized in Table 4.4. As expected (due to the difference in training conditions), the TIMIT–trained classifiers' performance is considerably worse on the machine segmentations than on the hand–labeled segmentations. As can be seen from the third and fourth columns in Table 4.4, adding machine–generated segmentations to the set of training vectors results in improved classification accuracies for the TIMIT and Machine testing conditions for all three classifiers.

It should be noted that the increase in the number of training vectors (from 140,179 phonetic training vectors for the TIMIT segmentations to the combination using 280,416 vectors) is achieved using the same amount of speech (NIST–Train). However, a large

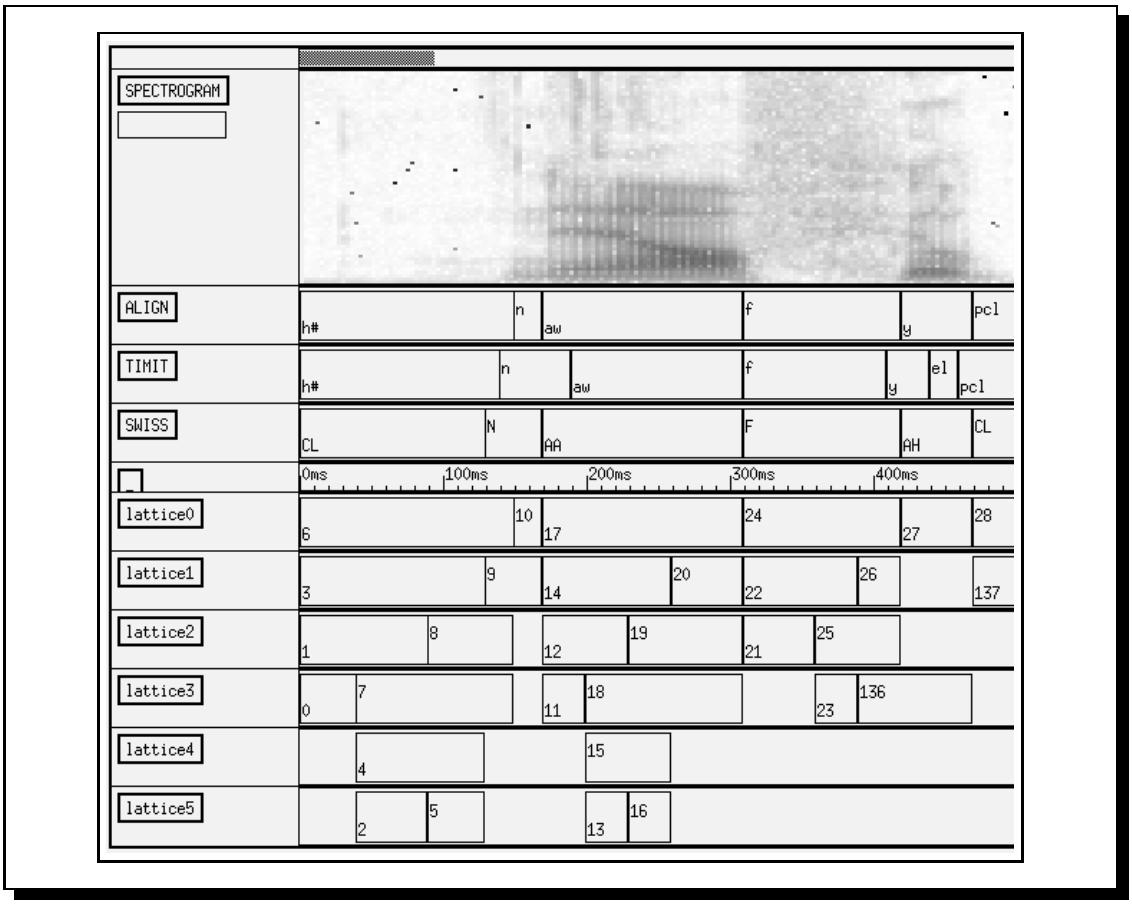

Figure 4.2: Example for positive and negative training segments. The figure shows (top to bottom): spectrogram, automatic alignment (ALIGN), hand–labeled transcription (TIMIT), phonetic recognition (SWISS), and five levels of the segment lattice (lattice1–5).

portion of the additional machine–generated segments are identical to the original hand–labeled segments and hence produce identical training vectors. No attempt was made to filter out duplicate training examples.

4.4 Training on Errors

Neural networks, which are the classifiers of choice in this work, are discriminantly trained which will allow us to use false–positive and false–negative examples, generated by the phonetic search, during training to improve the recognition (not necessarily the classification) performance. That is, we can in principle further improve the similarity between training

Table 4.4: Classification Results using Machine Segmentations, trained on NIST–Train and tested on MIT–Dev. The final two columns correspond to different testing conditions: either hand–generated segments (TIMIT) or machine–generated segments (MACHINE) were used.

Machine–Segmentation Classification Results

Classifier	Segmentation Method	Accuracy	
		TIMIT	MACHINE
Phoneme	TIMIT	74.1%	70.4%
Phoneme	TIMIT + Machine	**75.1%**	**72.8%**
Consonant	TIMIT	78.8%	76.0%
Consonant	TIMIT + Machine	**79.2%**	**78.0%**
Vowel	TIMIT	71.6%	67.4%
Vowel	TIMIT + Machine	**72.7%**	**70.3%**

and recognition conditions by including in the training sets not only those segments which correspond to the true cases of the phonemes, but also the wrong segmentations which the system will have to reject (via low posterior probabilities for all phonemes of that particular segment) during recognition. Figure 4.2 shows the two types of recognition errors that can be used in a "train–on–error" scheme: the initial nasal /n/ (lattice segment 9) is an example for a false positive, whereas the semi–vowel /y/ is misclassified as /ah/.

Our initial experiments indicate however, that finding the correct balance between error vectors (false positives and misclassifications) and true positive examples for training is difficult. Additionally, the confusability between the true phonetic categories and the newly created "Not–A–Phoneme" category is high, thus further confounding the training problem. We believe the idea of training the classifier on (segmentation and recognition) errors to be an important one but we will have to defer a more detailed investigation.

4.5 Summary

In this chapter, we reported on the segment classification experiments using cepstral–based features. Table 4.4 contains a summary of the various classifiers trained and tested on either TIMIT or machine–generated segmentations. Adding the machine–generated segmentations led in all cases to a performance improvement on both the TIMIT and the machine–segmentation based test conditions. The overall performance of these classifiers are comparable to, albeit slightly inferior to, the best results reported in literature (see Table 2.3). We will use the classifiers constructed in this section in phonetic recognition experiments described in Chapter 7.

Chapter 5

Robust, N–best Formant Tracking

In this thesis, we propose to use formant features to explicitly model the dynamics of speech. The eventual goal is to improve the accuracy of phonetic recognition in a segment–based speech recognition system. The success of this approach depends critically on the performance of the formant tracker, the algorithm which extracts formant frequency information (from now on called "formants").

Estimating the formants based on short–term spectral analysis is straightforward if the local information is pronounced (e.g. [34]). However, in practice we often have to deal with a flat short–time spectrum, where the information regarding the location of the formants can only be reconstructed from context by "tracking" the formants. Taking a global view allows the algorithm to compensate for incomplete or ambiguous local information.

As mentioned in Chapter 1, the idea of using formant features for speech recognition has been around for a long time, and despite numerous attempts, nobody (to the best of our knowledge) has been able to devise a formant–tracking algorithm which performs with sufficient accuracy to compete with cepstrum–based speech recognizers. However, we have observed that in virtually all sonorant segments, only a few consistent interpretations of formants are possible. Therefore we propose to extend the formant–tracking paradigm to find the N best interpretations rather than the single best as has been done in the past ([47, 83, 84]). The N best interpretations are subsequently presented to a phonetic segment classifier. The results of these classifications are then used by a standard phonetic search using language constraints (e.g. bigram) to find the single best interpretation, thus overcoming the lack of robustness of traditional formant trackers by delaying the final decision until after phonemic classification.

The formant–tracking algorithm assumes that the speech is already pre–segmented into sonorant, obstruent, nasal and silence segments (see Chapter 3 for details). The tracking algorithm is only applied to sonorant regions hence avoiding tracking problems across sonorant / obstruent boundaries as reported by Talkin [83].

The formant tracker uses a two–pass algorithm, similar to the work done by Laprie et al. ([47, 48]). The first pass finds individual formant tracks or parts thereof, called *elementary tracks* (Section 5.1). The second pass combines the elementary track information into consistent interpretations of F1, F2 and F3 using a wild–card mechanism to cope with misleading cues from the signal processing, such as insertions and deletions of parts of formants (Section 5.2). Since there might be more than one consistent interpretation of formants, the search produces a list of the N best interpretations.

After describing the algorithm in Sections 5.1 and 5.2 we will evaluate the quality of the proposed algorithm.

5.1 Finding Elementary Tracks

There are three popular mechanisms for generating formant candidates for a given sonorant frame:

1. computing the complex roots of a linear predictor polynomial [2],
2. peak picking of a short–time spectral representation [77],
3. analysis by synthesis [66].

Recently, Welling and Ney [89] proposed a formant estimation method based on digital resonators (see Section 8.2.1 for a possible application of this idea in the context of this work).

In this work we have chosen to pick peaks of a 20–th order LPC spectrum to generate formant candidates for each sonorant frame of speech. We chose a high LPC model order to generally generate more formant candidates per frame. The spectrum is discretized linearly into 32 frequency bands. Therefore, the formant location is the index of the frequency band.

Algorithm for Finding Elementary Tracks

1. Compute the 20–th order LPC and convert it into a 64–point power spectrum for each sonorant frame.
2. For each sonorant frame, find the local maxima in the lower 32 power–spectrum values (0 – 4kHz).
3. Initialize the elementary track hypotheses with the local maxima of the first frame of the sonorant segment.
4. For every active hypothesis, connect the hypothesis to the closest unused maximal point. If there is no such connection then terminate this hypothesis.
5. Add all unused maximal points to the set of hypotheses.
6. Repeat steps 4 and 5 for the next frame until the end of the sonorant region is reached.

Figure 5.1: Elementary Track Finding Algorithm

The algorithm that finds elementary tracks is summarized in Figure 5.1 and an example of the elementary tracks generated can be seen in the second display of Figure 5.6.

The start or the termination of an elementary track (because of an unused maximal point or lack of a connector) defines a division of the sonorant region into *sub–segments*. These sub–segments define the step size of the dynamic–programming search (described below).

The example of the word "brother" in Figure 5.6 shows that the LPC algorithm cannot resolve the apparent formant merge of F2 and F3 in the beginning and the end of the phoneme /r/ (partly due to the low energy of F3) given the chosen model order. The same phenomenon happens at the end of the retroflexed schwa /axr/. As can be seen in the bottom display of Figure 5.6, the search will be able to "correct" those inaccuracies of the signal processing step by using a wild–card mechanism in a search, based on dynamic–programming principles, described next.

5.2 N–best Search

The search for consistent formant interpretations (joint search for F1, F2 and F3) is a dynamic–programming algorithm similar to the Viterbi search. The search nodes contain the current score and pointers to the elementary tracks representing the formant frequencies $F_i \quad (i = 1, 2, 3)$. The track elements pointed to in turn store information about the trajectory of the track, more precisely the location for the track at each frame of the sub–segment. In order to make the search algorithm more robust to errors of the previous processing stages, a wild–card mechanism was implemented (see below). A wild card acts similarly to a real elementary track with the difference that there is no underlying track behavior other than the knowledge about the track location in the beginning of the sub–segment.

The search algorithm is summarized in Figures 5.2 and 5.3. As mentioned above, the search progresses on a sub–segment by sub–segment basis.

N–best Search Algorithm

1. Initialize the search by hypothesizing sets of F1, F2 and F3 locations using the elementary tracks of the initial sub–segment and additional *initialization rules*.
2. Apply the *consistency rules* to those hypotheses to get an initial score for each.
3. Expand each hypothesis with elementary tracks of the next sub–segment using the *expansion rules*.
4. Apply the consistency rules to those extensions and update the scores.
5. Repeat steps 3 and 4 for all sub–segments until the end of the sonorant region is reached.

Figure 5.2: N–best Search Algorithm

Generally, formant–tracking algorithms try to find a good trade–off between maximizing the amount of energy "explained" (sum of formant amplitudes) by a given interpretation and some sort of smoothness constraint (e.g. [48], [83]). Our goal is to find *consistent* interpretations of the formant information as represented by the elementary

Initialization Rules

1. Add all elementary tracks of the first sub–segment which are within the expected range for the formant F_i to the set of candidates S_i.
2. If there exists a second sub–segment in the tracking region, then add a *wild card* to S_i for every elementary track in this second sub–segment. The beginning locations of the wild cards are set to the locations of the elementary tracks in the second sub–segment.
3. If S_i is empty, then add a wild card to S_i with the default location for F_i.

Figure 5.3: Initialization Rules for the Search

tracks. Therefore, we "count" the number of *consistency violations* as well as the number of *preferred behaviors* of a hypothesis. The list of *credits* is shown in Figure 5.4. A *natural connection* is a continuation of an elementary track of one sub–segment in the next sub–segment (given that the track was not the cause for the sub–segment boundary). The parameter $MAXDELTA$ controls the penalty (negative credit) for considering a large difference in formant locations across a sub–segment boundary. The total credit for an extension is then multiplied by the number of frames contained in the sub–segment and added to the current score for the interpretation.

Credit Assignment Rules

$+1$	for following a *natural connection* (not applicable at the beginning of the search)
$+1$	for using a unique elementary track (versus using a wild card)
0	for using a wild card
-2	for not using an elementary track of lower frequency
$-k$	for connecting 2 tracks further than $k * MAXDELTA$ apart

Figure 5.4: The Credit Assignment Rules for Computing the Score of a Hypothesis.

The expansion rules (Figure 5.5) generate reasonable extensions for a given hypothesis. The rules are designed to apply to each hypothesized formant location separately, independent of the location of the other (two) formants. The parameter $MAXJUMP$ controls the maximum allowable difference between the end location of the current formant hypothesis and the elementary track to connect to. Because the search is advanced by one sub–segment at a time and the expansion rules are concise, the algorithm works efficiently enough to eliminate the need for a special pruning algorithm.

Expansion Rules

1. Add all elementary tracks which are within $MAXJUMP$ bands of location of F_i to S_i.
2. If S_i is empty, add a wild card with the same location as the last know hypothesis to S_i.
3. Discard all extensions which violate the ordering constraint on the formant locations:

   location of $F_1 \leq$ location of $F_2 \leq$ location of F_3.

Figure 5.5: Expansion Rules for the Search

An example of the search output is shown in the bottom display of Figure 5.6. The lighter grey color indicates the presence of a wild card as part of the best–scoring hypothesis. In a post–processing step, the wild–card locations are replaced by a linear interpolation between the last known points (indicated by a dashed line in Figure 5.6). As can be seen in this example, the wild–card mechanism was able to cope with the formant merges in the retroflexed parts of both sonorant regions tracked. Figure 5.7 further shows the second and third solution found by the formant tracking algorithm for the example word "brother".

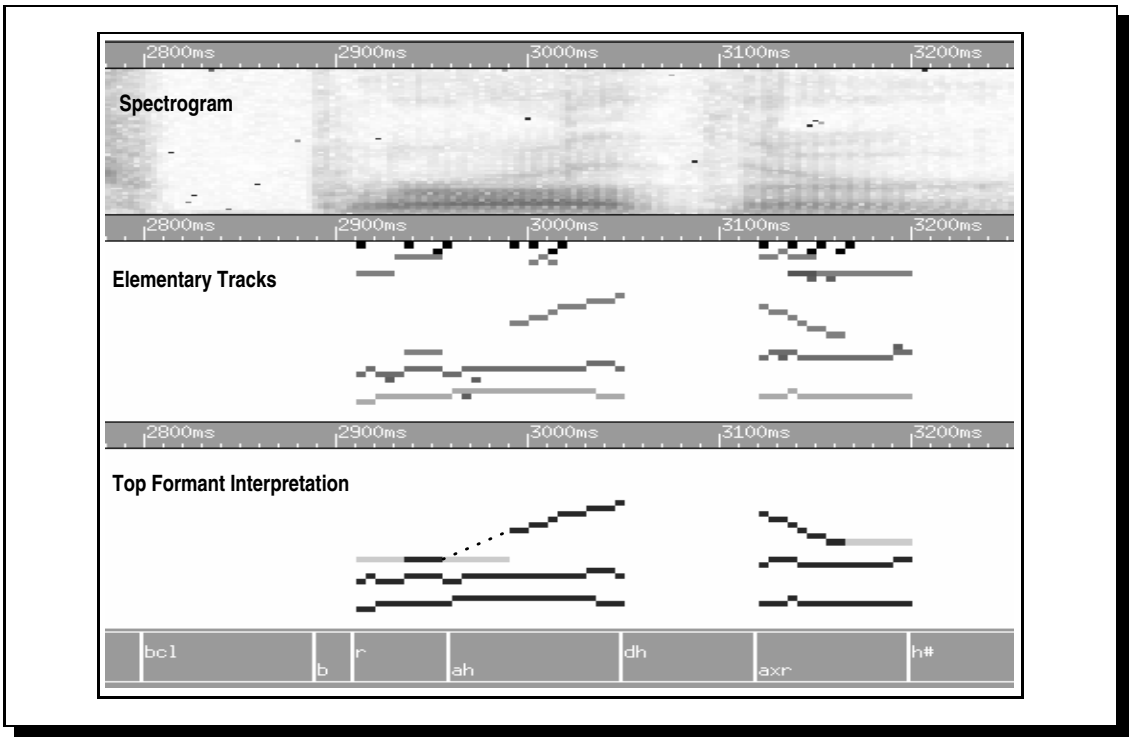

Figure 5.6: Display of (top to bottom): Spectrogram (0 – 8kHz), Elementary Tracks (0 – 4kHz), Best Search Hypothesis (dashed line indicates the wild card interpolation), Phonetic Transcription of the word "brother".

5.3 Evaluation

Because the role of our formant tracker is simply to pass N reasonable hypotheses to the segment classifier, we have restricted ourselves to a few informal tests of our formant tracker. The real tests lie in the application of the tracking algorithm in segment classification and phonetic recognition described in Chapters 6 and 7.

In a preliminary study [78], we evaluated the formant tracker on 10 test utterances from the TIMIT database containing 148 sonorant segments (vowels, liquids and glides) for a total of 101 sonorant regions. The analysis shows that for all but 3 segments the best scoring hypothesis was judged correct[1], with the remaining 3 cases being ranked second

[1]The correctness was judged by the author by comparing the solution to the ESPS formant tracks and inferring the expected correct location from the knowledge of the phonetic labels.

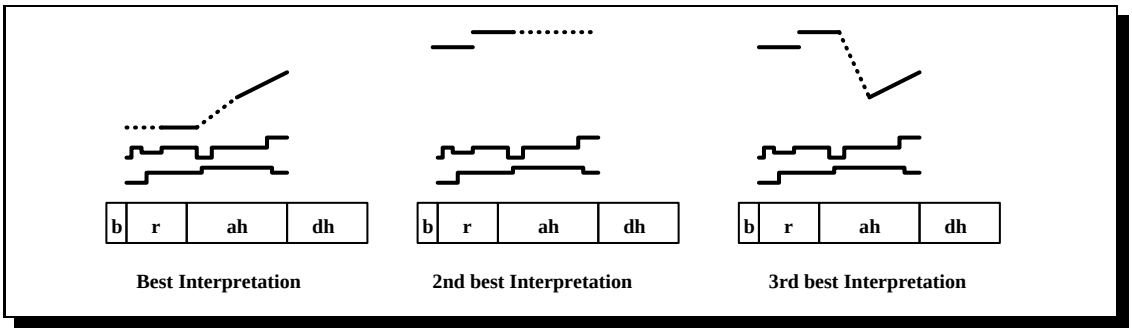

Figure 5.7: Top 3 Formant Interpretations of the word "brother" (as in Figure 5.6). The dashed lines are indicating the use of wild cards.

in the N–best list.

Due to the lack of a database with labeled formant information, we restrict ourselves at this point to analyze the diversity of the N–best list. For this purpose, we applied the formant tracker to all the sonorant regions of 3698 utterances of the training set, producing a 10–best list for each region. For each phonetic segment contained within that region we measured the diversity by counting the number of identical solutions among the top N interpretations. The results are tabulated for various values of N in Table 5.1.

Table 5.1: Diversity of the N–best Formant Tracker. For example, 12.2% of all vowel and semi–vowel segments had exactly 2 unique formant interpretations.

Diversity of N–best List

N	1	2	3	4	5
Diversity	48.1%	12.2%	6.1%	4.9%	4.3%
N	6	7	8	9	$\geq$ 10
Diversity	4.3%	4.0%	3.9%	3.7%	8.6%

The relatively high number of identical solutions among the top N interpretations has two causes. Firstly, due to the fact that the formant tracker was applied to an entire sonorant region instead of a single segment, there are N interpretations per region, not segment, and therefore less diversity per segment is seen. Secondly, it also suggests that

a more efficient pruning mechanism for the algorithm is needed.

5.4 Summary

In this chapter, we have described a new robust, N–best formant tracker. This 2–pass algorithm first defines elementary tracks which are subsequently combined into N consistent interpretations of the formant information using a dynamic–programming search with a wild–card mechanism. The primary objective of the search is not to maximize the energy explained by a single hypothesis. The goal of this formant tracker is to find up to N consistent interpretations which are then passed on to the next processing step, the segment classification. This proposed algorithm gains its robustness from using wild cards to overcome limitations of the earlier processing steps and from the fact that more than one alternative is retained.

In this thesis, we will use the formant–tracking algorithm to produce dynamic features for vowel classification described next in Chapter 6. Other applications for formant tracking information include:

- parameter extraction for a text–to–speech system
- analysis component for a formant based vocoder
- speaker identification, adaptation or normalization

Chapter 6

Explicit Formant Features for Segment Classification

After describing segment classification experiments using cepstral features in Chapter 4, we will now examine feature sets based on explicit formant features and other knowledge–based features. The experimental conditions (training/test sets) are identical to the previous vowel/semi–vowel classification experiments.

This chapter describes one of the key contributions of this thesis research: vowel classification using explicit formant features in an N–best framework. Before we explain the basic trajectory approximation methods investigated in this thesis, piecewise–linear regression (Section 6.2) and Legendre polynomials (Section 6.3), we describe knowledge–based features in Section 6.1 that will be used in conjunction with the features capturing the formant trajectories to build vowel/semi–vowel classifiers. Finally, in Section 6.4, we describe the N–best classification framework.

6.1 Knowledge–Based Features

Perceptual experiments using natural and synthetic vowels (see Section 2.3 for an overview of some of the literature) suggest that the following features are important for identification: intrinsic vowel duration, fundamental frequency, relative duration of onglides and offglides (for discrimination of tense and lax vowels). Additionally, we also included formant amplitude and formant bandwidth in our experiments, despite the fact that Klatt has concluded from his experiments [40] that these features are less important than formant location.

The classification performance including these additional features in combination with either basic approximation (line segments, Legendre polynomials) are summarized in Tables 6.1 and 6.3. The classifiers were trained on the NIST training set and tested on the MIT development test set. The performance of the MFCC–feature based classifier (see Section 4.2) is provided as a point of reference.

Pitch Feature

In this work, we use a pitch tracker based on the cepstrum method of pitch extraction. This tracker, developed by Van Vuuren, [86]) allows the analysis window length to be changed adaptively for greater dynamic range. We use a 10ms increment and allow a window size of 10–30ms. The pitch track of each utterance is smoothed by a 5–point median filter. The pitch feature is defined as the the averaged pitch estimates over the middle 80% of the segment.

Onglides and Offglides (F1 Intervals)

As suggested by Peterson and Lehiste [69], symmetry and duration of the onglides (formant transition into the vowel nucleus) and offglides of F1 in particular are important features for the tense/lax vowel separation. We measure the relative distance of the end of first line segment from the beginning of the segment, as well as the beginning of the last line segment from the end of the phonetic segment. This feature was only used in conjunction with the line–segment baseline features, since an interpretation in the context of Legendre polynomials is not directly available.

Formant Amplitude and Bandwidth

It has been argued by Bladon et al. [7], among others, that the formant locations are not sufficient for vowel perception. They argue that capturing the spectral shape is essential in describing the nature of the vowel sound. To investigate this possibility, we used the formant amplitudes (energy at formant locations for the middle of the segment) and formant bandwidth (distance between 3db points) as additional features. Again, the individual feature measurements are averaged over the middle 80% of the segment. Even

Piecewise Linear Regression

1. Initialize the algorithm by dividing the track into k equally long segments.
2. Compute slopes and intercepts for each segment given the current segmentation.
3. Using a dynamic programming search, find the segmentation which minimizes the overall regression error given the current slopes and intercepts.
4. Repeat Step 2 until either the segment boundary locations are no longer changing or until the regression error is below a threshold.

Figure 6.1: Piecewise Linear Regression Algorithm

though these features are redundant with one another in a cascaded formulation of vowel production [67], we find them to contribute independently in our system.

6.2 Piecewise–Linear Regression

A variety of methods to capture the dynamic shape of trajectories in the parameter space have been proposed in the past. The literature review in Chapter 2 explains some of the methods used by other researchers to approximate the trajectories of cepstral or spectral features. In this work, we use the Piecewise Linear Regression (PLR) algorithm proposed by Krishnan and Rao [44] to approximate each formant track by three line segments. The PLR algorithm, summarized in Figure 6.1 generally converges to the optimal solution within 3 to 5 iterations. In practice, we have found the computational load to be negligible. The boundary locations for the line segments are chosen independently for each formant track.

Figure 6.2 illustrates the application of the PLR algorithm. The second formant of the phoneme /ay/ has been approximated by 3 line segments (the line segments are extended beyond the connection points for better visualization).

The features for the vowel segments are determined by measuring the start and end points of the line segments for each formant (marked by dots in Figure 6.2). Additionally,

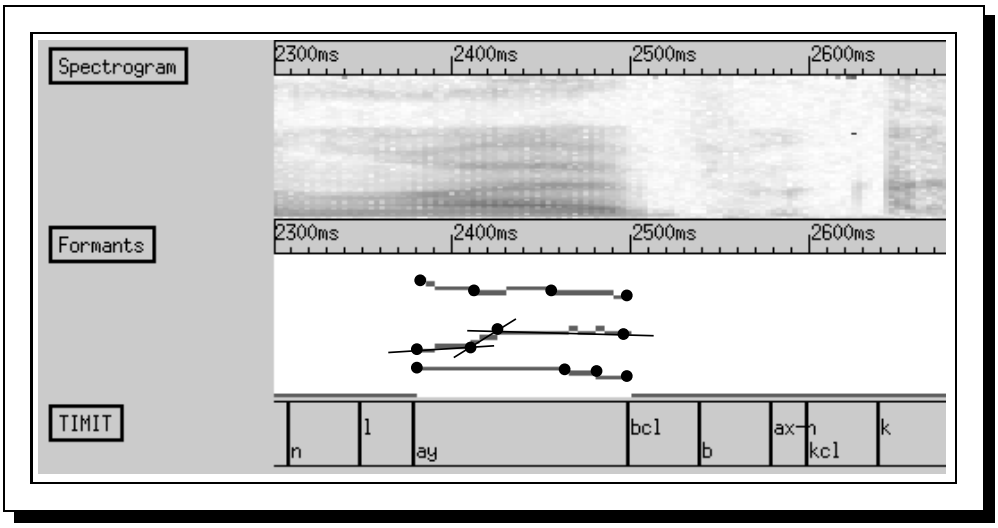

Figure 6.2: Piece-wise Linear Regression Formant Features: the second formant of the phoneme /ay/ is approximated by 3 line segments using the PLR algorithm. The dots indicate the locations of the formant features for this particular segment.

if the neighboring segments also contain formant information we add the locations of the last and first line segments of the left and right adjacent segments respectively. After adding the segment duration to the list of features, we end up with 19 features describing the dynamics of the segment (see Figure 6.3).

In choosing to approximate the formant trajectories by three line segments, we intended to create a "direct" feature representation, providing the classifier with features describing the important information directly, rather than encoded, as is the case for polynomial approximations (see the next section for such a polynomial fit). The vowel/semi–vowel classification results using the PLR features in conjunction with knowledge–based features described in the previous section are reported in Table 6.1.

It should be noted that the amount of contextual information is limited to the case of adjacent sonorant segments, whereas the cepstral–based classifiers are always given contextual information. This might be a contributing factor to the considerably worse classification performance for PLR features (66.0%) compared to the MFCC–based classifier (71.6% and 73.1%) as reported in Table 6.1.

PLR Formant Features

1. For each formant:
   - left (right) end point of the first (third) line segment,
   - connection points (averaged) between the first and second resp. second and third line segments,
   - If there is a sonorant segment to the left (right) of the current phonetic segment:
     - first (last) point of the last (first) line segment of the prior (next) phonetic segment

     otherwise
     - default value
2. Segment duration

Figure 6.3: Piecewise–Linear Regression Formant Features for Sonorant Segments

Table 6.1: Vowel Classification Results For PLR Feature Sets. The codes for the knowledge–based features (Section 6.1 are: Duration (D), Amplitude (A), Pitch (P), Bandwidth (B), and F1 Intervals (I)

PLR Vowel/Semi–Vowel Classification Results

Feature Set	# Features	Accuracy
PLR (3 Line Segments) + D	19	**66.0%**
PLR + D + A	22	68.1%
PLR + D + P	20	67.9%
PLR + D + B	22	66.5%
PLR + D + I	21	67.2%
PLR + D + A + P	23	**68.2%**
MFCC + D	113	**71.6%**
MFCC + D + P	114	**73.1%**

6.3 Legendre Polynomials

In this section we describe an alternative method for approximating the trajectory of the formant tracks using third–order Legendre polynomials (LEG). We chose the Legendre polynomials because they constitute an orthonormal basis set, which results in more robust parameter estimations. The third–order polynomials in Table 6.2 are taken from [12], where M is number of formant trajectory data points $\{y_1, \ldots, y_M\}$ available for the parameter estimation. The Legendre coefficients a_j are computed as follows:

$$a_j = \frac{1}{M+1} \sum_{i=0}^{M} y_i \cdot \phi_j(\frac{i}{M}) \quad .$$

An arbitrary point $x \in (0,1)$ within the scaled segment can be reconstructed by:

$$f(x) = \sum_{j=0}^{3} a_j \cdot \phi_j(x) \quad , \quad x \in (0,1) \quad .$$

The average prediction error can then be computed as:

$$PE = \frac{1}{M+1} \sum_{i=0}^{M} \mid y_i - f(\frac{1}{M}) \mid \quad .$$

As can be seen from Table 6.3, using the Legendre coefficients a_j together with the average prediction error per formant trajectory significantly improves the classification performance from 66.0% for the PLR features to 69.6%.

From the vowel/semi–vowel classification results in Table 6.3 it can be seen that the combination of Legendre coefficients and knowledge–based features perform virtually identically to the cepstral features without pitch information. However, if we add the pitch feature to the cepstral features, we can improve the classification performance to 73.1%. The combination of MFCC features with the Legendre approximation and knowledge–based features (including pitch) cannot further improve the classification performance significantly. This result reflects the fact that there is a high degree of similarity between formant–based features and cepstral features. Additionally, TIMIT contains high–quality read speech, for which pitch extraction is easier than for spontaneous, telephone–quality speech. As indicated earlier (Chapter 1), the advantages of formant–based featrures are in their use of global constraints in more appropriate feature space for principled speaker normalization.

Table 6.2: Legendre Polynomials

Bases for Legendre Polynomials

$$\phi_0(x) = 1$$

$$\phi_1(x) = \left[\frac{12M}{(M+2)}\right]^{1/2} \cdot \left[x - \frac{1}{2}\right]$$

$$\phi_2(x) = \left[\frac{180M^3}{(M-1)(M+2)(M+3)}\right]^{1/2} \cdot \left[x^2 - x + \frac{M-1}{6M}\right]$$

$$\phi_3(x) = \left[\frac{2800M}{(M-1)(M-2)(M+2)(M+3)(M+4)}\right]^{1/2} \cdot \left[x^3 - \frac{3}{2}x^2 + \frac{6M^2 - 3M + 2}{10M^2}x - \frac{(M-1)(M-2)}{20M^2}\right]$$

$$x \in (0,1) \qquad M \geq 3$$

Table 6.3: Vowel Classification Results For LEG Feature Sets

LEG Vowel/Semi–Vowel Classification Results

Feature Set	# Features	Accuracy
LEG + PE + D	22	**69.6%**
LEG + PE + D + A	25	70.6%
LEG + PE + D + A + P	26	71.3%
LEG + PE + D + A + P + B	29	**71.8%**
MFCC + D	113	**71.6%**
MFCC + D + P	114	**73.1%**
MFCC + LEG + PE + D + A + P + B	141	**73.4%**

As in Chapter 4, we augmented the training vectors with examples based on machine–generated segmentations. Table 6.4 summarizes the results.

Table 6.4: Vowel/Semi–Vowel classification results using classifiers trained on TIMIT and machine–generated segmentations

Classification Results Using Machine–Generated Segmentations

Features	Segmentation Method	Accuracy	
		TIMIT	MACHINE
LEG + MFCC	TIMIT	73.4%	70.8%
LEG + MFCC	TIMIT + Machine	**74.1%**	**71.7%**

6.4 N–best Classification Paradigm

The formant tracker described in Chapter 5 produces a list of N consistent interpretations. The decision on the correct interpretation is delayed until after the classification stage. Because we are interested in phonetic recognition, we do not attempt to determine whether or not the correct interpretation was found. Our objective is to classify the segments in question correctly. Therefore, each interpretation is used to estimate phonetic classification probabilities using a neural network classifier. The phonetic category a^* with the highest probability over all N interpretations $Interp_k$ is used to label the segment:

$$a^* = \underset{a,k}{\operatorname{argmax}}\, p(a|Interp_k) \qquad \text{for k = 1 ... N} \tag{6.1}$$

In essence, the question is: "Assuming this is the correct formant interpretation, which phoneme would it be?" Note that we propose to use more than one set of features per segment in the classification process! This is a novel concept, as far as we know.

Unfortunately, we do not have access to previously–labeled formant data (for the TIMIT database) for training and testing purposes. Therefore, instead of labeling the

entire training set by hand, we elected to employ an iterative training procedure described in Figure 6.4.

Iterative Training Procedure

1. Train an initial set of segment classifiers (with different numbers of hidden nodes and weight initializations) assuming that the highest scoring formant interpretation is the correct one.
2. Choose the classifier among the newly trained ones which performs best on development test set 1.
3. Use the classifier from Step 2 to select the interpretation with the highest classification probability for the correct category for the training set and development test set 2.
4. Train a new set of segment classifiers.
5. Repeat Steps 2 through 4 with reversed roles for the two development test sets until no further improvement in the overall classification accuracy can be found.

Figure 6.4: Iterative Training Procedure for Sonorant Segment Classifier

The experiments using the iterative training procedure were conducted using the OGI–Train, OGI–Dev1, and OGI–Dev2 training and test sets (see Table 4.1) because we needed multiple, independent development test sets. As a consequence of the smaller initial training set (OGI–Train), our initial classifier only achieved 63.4% classification accuracy.

Table 6.5 summarizes the results of the iterative training process. The number of formant interpretations per segment was set to 3. The iterative process reduced the error rate by 2.5%. This small decrease is mainly due to the fact that most of the interpretations are actually identical (see Section 5.3) therefore reducing the choice for the selection of training patterns. An improvement of the pruning strategy of the formant tracker (see Chapter 8) should lead to a higher performance gain for the iterative training procedure.

Unfortunately, we were unable to achieve any improvement with $N > 1$ when starting with the best classifier ($N = 1$), trained on the entire TIMIT training set NIST–Train, and using $N = 5$. See the discussion at the end of this section for a possible explanation

Table 6.5: Classification Results for Iterative Training Procedure

Classification Results: Iterative Training

Iteration	Training Sets	Test Set	Accuracy
Init	OGI–Train	OGI–Dev1	63.4%
		OGI–Dev2	62.7%
1	OGI–Train OGI–Dev2	OGI–Dev1	63.6%
2	OGI–Train OGI–Dev1	OGI–Dev2	63.6%
3	OGI–Train OGI–Dev2	OGI–Dev1	63.8%
4	OGI–Train OGI–Dev1	OGI–Dev2	63.8%
5	OGI–Train OGI–Dev2	OGI–Dev1	**64.3**%
6	OGI–Train OGI–Dev1	OGI–Dev2	63.6%

of this negative result.

"Oracle Experiment"

As mentioned above (Equation 6.1), the correctness of the classification in the above experiments was measured by comparing the label of the category with the highest classification probability of all the formant interpretations with the TIMIT transcriptions. To check how often an interpretation of lower classification score corresponded with the true TIMIT label, we evaluated the performance of the best iteratively–trained vowel classifier under the assumption that a classification is correct, if for any of the N interpretation the segment was classified correctly:

$$\exists k \quad a^* \;=\; \underset{a,k}{\operatorname{argmax}}\, p(a|Interp_k) \quad \text{AND} \quad a^* \text{ correct classification} \tag{6.2}$$

Table 6.6 summarizes the results of this analysis where we have used the PLR features along with the segment duration for a total of 19 features to classify the vowels and semi–vowels of the OGI–Dev1 set. It shows, that if we had an "*Oracle*" inside the recognizer,

which could determine the correct interpretation, we could improve the classification accuracy from 71.8% to 76.3% which corresponds to an error reduction of 17%! This result suggests that $k > 1$ is often the correct interpretation.

Table 6.6: Classification Results for Oracle Experiment

Classification Results: *Oracle Experiment*

N	1	2	3	4	5
Classification Accuracy	71.8	73.7	74.7	75.2	75.6
Different Classifications	1.00	1.09	1.15	1.19	1.23
N	6	7	8	9	10
Classification Accuracy	75.9	76.1	76.2	76.3	**76.3**
Different Classifications	1.25	1.27	1.28	1.28	1.29

The potential performance improvement for the N–best paradigm could also be due to the fact that up to N different classification results are compared to the correct label, which naturally increases the chance of producing a correct classification. In order to be able to rule out this possibility, we counted the number of different classification answers proposed for each N. The results in row 3 of Table 6.6 indicate that only in selected cases the subsequent classifications were different from the answer for $N = 1$. This result suggests that the N–best paradigm has considerable potential. The problem of not being able to improve on the $N = 1$ case (see above) might hinge on the fact that the outputs of the MLP classifier are not strictly comparable to each other since they are estimated using different feature vectors (only a comparison among the categories for the same feature vector are permissible). In theory, the outputs of the MLP classifiers are posterior probabilities [32] and should be comparable, but in practice the assumptions regarding size of training data and classifier resources are not met and hence the outputs are not true posterior probabilities. Future research into the rescaling of MLP outputs is needed to solve this problem properly. Initial experiments using MLPs for a recombination of either the classifier outputs or presenting all N feature representation to a classifier have not

resulted in any improvement over the $N = 1$ case. Additionally, the experiment suggests that the relatively small number of training examples poses a problem for the iterative training procedure.

6.5 Summary

In this chapter, we have created vowel/semi–vowel classifiers that used features based on the N–best formant tracker described in Chapter 5. When approximating the formant trajectories by third order Legendre polynomials and adding formant amplitude, formant bandwidth, as well as pitch and segment duration, we can achieve a comparable level of classification accuracy (71.8%) to the MFC–based classifier (71.6%) developed in Chapter 4 for $N = 1$, using only 29, albeit expensive (in terms of computation requirements) features compared to the 113 features for the MFC–based classifier. The best classification results are achieved by combining the MFCC and formant features yielding a 73.4% accuracy, which were further improved by adding machine–segmented training examples to 74.1%.

Subsequent experiments with larger Ns show only a small improvement. This is most likely due to the improper estimation of posterior probabilities by the MLP classifiers across different feature vectors. When ignoring the level of the classification scores (and only considering the classification result), we notice that there is a considerable potential for improvement (up to 76.3% classification accuracy) if the scaling problem can be solved.

The experiments in this chapter demonstrate that satisfactory classification performances can be achieved using formant–based features, despite the shortcomings of the current implementation of the formant tracker. The feature configuration is an initial attempt at exploring the power of formant–based features. We hope that future improvements will add additional credibility to the idea of using explicit formant features for classification. Additionally, similar improvements can be anticipated for consonant classification.

Chapter 7

Phonetic Recognition

In this chapter, we report on the phonetic recognition experiments using the segment lattice described in Chapter 3 and the segment classifiers developed in Chapters 4 and 6. We concentrate in this thesis on phonetic recognition experiments without the use of phonotactic constraints. While using a language model can significantly improve the recognition accuracy, it can also hide weaknesses of the recognizer.

First we introduce the probabilistic framework that underlies the search algorithm. Next, in Section 7.2, we discuss some implementation issues, i.e. modifications to the segment lattice and the basic search formula. Finally, the recognition results are reported and discussed.

7.1 Probabilistic Framework

The objective of phonetic recognition is to find the most likely sequence of phonetic units $A^* = \{a_1, a_2, \ldots, a_N\}$ given the acoustic observations $X = \{x_1, x_2, \ldots, x_T\}$:

$$A^* = \underset{A}{\operatorname{argmax}}\, p(A|X) \tag{7.1}$$

In the case of a segment–based framework, Equation 7.1 has to take into account all possible segmentations S of X:

$$A^* = \underset{A}{\operatorname{argmax}} \sum_{S_i \in \mathcal{S}} p(A, S_i|X) \tag{7.2}$$

For the sake of efficiency, most speech recognition systems identify the most likely state sequence. Therefore, Equation 7.2 is simplified to find the most likely joint segmentation and labeling:

$$A^* = \underset{A,S_i}{\operatorname{argmax}}\, p(A, S_i|X) \tag{7.3}$$

Because our segment classifiers (MLP) estimate posterior probabilities, we can expand Equation 7.3 as follows (analogous to the expansion of Leung et al. [55]):

$$A^* = \underset{A,S_i}{\operatorname{argmax}}\ \underbrace{p(A|S_i, X)}_{\substack{phonetic \\ classification}} \cdot \underbrace{p(S_i|X)}_{segmentation} \quad . \tag{7.4}$$

The first term in Equation 7.4 represents the segment classification task. The second term, the probability of a segmentation, is generally computed as the product of the individual segment probabilities $p(s_i|X)$:

$$p(S|X) = \prod_{i}^{M} p(s_i|X) \quad . \tag{7.5}$$

The estimation of the segment probability, however, is difficult. Unlike in the case of the segment classification, we do not have an intuitive notion of a probability for a segmentation, much less a feature representation for training a probabilistic estimator. Therefore, this problem has received little attention in the research on segment–based systems to date. The term $p(S|X)$ or alternatively $p(S)$ is generally approximated by a global constant. This constant is used to control the insertion and deletion rates of a recognizer[1]. One notable exception are Leung et al. [54]. They propose to estimate the probability of a segment by multiplying the probabilities of the external and internal boundaries of a segment. They estimate the boundary probabilities using a separately trained MLP classifier.

[1]Insertions and deletions factor into the recognition accuracy score. See Section 7.3 for details.

In this work, we are using the same approach of multiplying acoustic segment (AS) boundary probabilities; that is, the probability of a segment $s_i = \{as_m, \ldots, as_n\}$ is:

$$p(s_i|X) \;=\; p(b_n|X, AS) \prod_{j=m}^{n-1} (1 - p(b_j|X, AS)) \qquad . \tag{7.6}$$

However, we are taking advantage of previously computed information (while constructing the segment lattice) rather than training a separate classifier. The probability of the boundary b_k between the acoustic segments as_k and as_{k+1} is computed as the difference in the broad category probability distributions, where NB is the number of broad categories β_i:

$$\begin{aligned} p(b_k|X, AS) &= p(b_k|as_k, as_{k+1}, X) \\ &= \frac{1}{2} \sum_{i=0}^{NB} \mid p(\beta_i|as_k, X) - p(\beta_i|as_{k+1}, X) \mid \end{aligned} \tag{7.7}$$

The sum in Equation 7.7 ranges from 0.0 (in the case of identical probability distributions for as_k and as_{k+1}) to 2.0 (in the case the two distributions are disjoint, indicating the presence of a phonetic boundary)[2]; this explains the factor $\frac{1}{2}$ in front of the sum.

7.2 Implementation Issues

The search for the optimal path (in the sense of Equation 7.2) through the segment lattice is implemented as a dynamic programming algorithm, similar to the Viterbi search algorithm [25] used in most speech recognition systems. The fundamental algorithm is modified to be used in a segment–based framework in the following way: instead of advancing the search horizon one frame at a time, the search hypotheses are updated at each boundary b_k. Each search hypothesis terminating at this boundary is extended with all segment hypotheses starting at this boundary.

[2]Ideally, there are 2 differences in the sum of size 1.0 for the two categories which are the unique winners in as_k and as_{k+1} respectively.

All of our recognition experiments will be in a context–independent mode, since all of our classifiers model context–independent units (phonemes). See Goldenthal [31] for a discussion of implementation issues related to context–dependent recognition.

The selection of the appropriate segment classifier in the case of specialized version for vowels and consonants is guided by the information attached to each segment by the lattice–generation algorithm. The segment–classification probability $p(a|s_i, X)$ in the basic search Formula 7.4 has to be modified to include the probability of the classifier $p(\mathcal{C})$:

$$p(a|s_i, X, C) \;=\; \prod_{j}^{N} p(a|s_i, X, C)\; p(C|s_i, X) \qquad . \tag{7.8}$$

Experiments will show that this does not introduce any degradation since the results are virtually identical for the cases of separate and single classifiers (using the same classification features).

Similarly, the sonorant regions used for the formant tracker are inferred from the lattice. The system enforces consistency of formant interpretations across adjacent segments within the same sonorant region. That is, each sonorant segment is not treated separately if it is adjacent to another sonorant segment.

While experimenting with the system, we have found two simple modifications that improve the recognition performance. Firstly, the relative weighting of the components of the basic search formula (Equation 7.4) can be modified by introducing a scaling factor θ. Secondly, the segment lattice can be augmented by inserting additional segments in places of likely omissions by the original lattice generation algorithm.

7.2.1 Modified Boundary Probabilities

While Equation 7.4 defines the best scoring path through the segment lattice, we have observed that often the probabilities in Equation 7.4 are not estimated properly and therefore lead to incorrect paths being found. Figure 7.1 illustrates this problem.

Assuming that in this case the hypothesis $H_2 = \{B, C\}$ is the correct path, it can lose to hypothesis $H_1 = \{A\}$ based on the simple fact that fewer terms are multiplied in

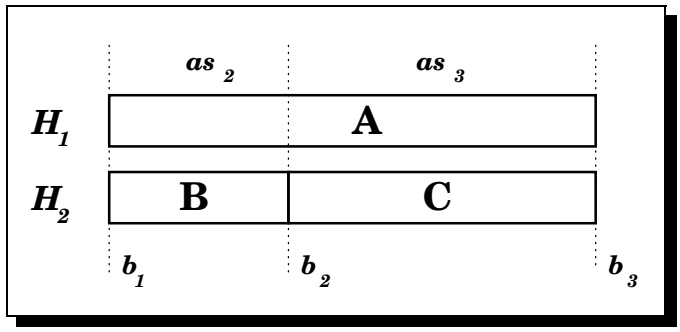

Figure 7.1: Search Hypotheses

determining the score S for each path:

$$\begin{aligned} S(H_1) &= p(a|A,X)\ p(A|X) \\ &= p(b_1)\ p(a|A,X)\ (1-p(b_2))\ p(b_3) \end{aligned}$$

$$\begin{aligned} S(H_2) &= p(b|B,X)\ p(B|X)\ p(c|C,X)\ p(C|X) \\ &= p(b_1)\ p(b|B,X)\ p(b_2)\ p(c|C,X)\ p(b_3) \end{aligned}$$

$$\begin{aligned} S(H_1) &< S(H_2) \\ &\Downarrow \\ p(a|A,X)\quad (1-p(b_2)) &< p(b|B,X)\ p(c|C,X)\quad p(b_2) \end{aligned} \tag{7.9}$$

From Equation 7.9 we can see that the probability of the middle boundary $p(b_2)$ plays a pivotal role. Even if the product of the segment classification probabilities for segments B and C are higher than $p(a|A,X)$, the boundary probability $p(b_2)$ still has to be larger than 0.5 for H_2 to win over H_1. In practice, we have observed that this is not always the case, especially for boundaries between two sonorant segments. We can try to correct this problem by rescaling the boundary probabilities; we have had success with a rescaling of the form:

$$p_s(b) = 1 - (1 - p(b))^3 \qquad .$$

Additionally, we can control the influence of the boundary probabilities on the search

path by introducing a scaling factor θ in Equation 7.4:

$$A^* = \underset{A,S_i}{\operatorname{argmax}} \prod_{s_j \in S_i} p(a|s_j, X)^{\theta} \cdot p(s_j|X)^{(1-\theta)} \tag{7.10}$$

7.2.2 Augmented Segment Lattice

Another source of recognition errors is the production of insertions in the case where only a single (incorrect) segment spans a region of speech. Systematically augmenting the segment lattice with combinations of the adjacent segments with the single segment gives the search an alternative. Figure 7.2 illustrates this process.

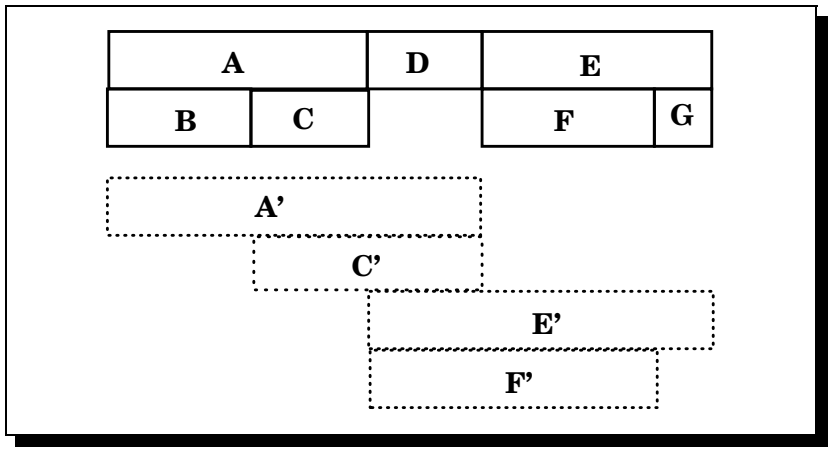

Figure 7.2: Augmentation of Segment Lattice due to single Segment D

In the original segment lattice (segments $A, \ldots, G$) contains a single segment D, which forces the addition of segments A', C', E', and F'. This process is applied to single short segments, and reduces the insertion rate as can be seen in Table 3.3.

7.3 Recognition Results

Evaluating the performance of a speech–recognition system is more complex than scoring a segment–classification experiment. The speech–recognition system generates a transcription which is compared against the reference transcription provided by the TIMIT

database. An alignment program[3] is used since in general the number of tokens in these two strings are different. When comparing the two transcriptions, we can observe three types of errors, summarized in Table 7.1.

Table 7.1: Types of Recognition Errors

Types of Recognition Errors	
Substitutions:	Segments in both transcriptions are lined up, but the labels are different
Insertions:	an additional segment is proposed by the search algorithm
Deletions:	a true segment is missed by the search algorithm

The scoring program finds the alignment which minimizes the sum of segment insertions, deletions, and substitutions over an utterance. However, since theoretically, one can always achieve 100% correct classification by generating all possible sequences, an additional measurement, the recognition accuracy, is generally used as the significant performance number (Table 7.2).

Table 7.2: Definitions of Performance Measurements

Performance Measurements

$$\textbf{Percent Correct} = \frac{\text{number of phones correct}}{\text{number of phones in the reference transcription}}$$

$$\textbf{Accuracy} = \text{Percent Correct} - \text{Percent Insertions}$$

The phonetic recognition experiment using different classifiers developed in Chapters 4 and 6 are summarized in Table 7.3. The language model is an implicit unigram, since the posterior probabilities estimated by the MLP classifiers contain the prior probabilities.

[3]NIST provides a reference implementation

The selection of the appropriate segment classifier in the case where separate vowel/semi–vowel and consonant classifiers were used (experiments C–G in Table 7.3) was based on the broad–phonetic tagging by the lattice–construction algorithm. Experiment G used the combination of formant and MFCC features. The context–independent, unigram recognition result obtained by Glass et al. [29] is provided as a point of reference.

Table 7.3: Phonetic recognition results for various classifiers using a unigram language model. "T" and "M" refer to the training segmentations obtained from either TIMIT or our automatic process respectively (see Section 4.3). The experiments are labeled A–H for easier reference in the text. Cons = Consonant. Fmt = Formant.

Phonetic Recognition Results: Unigram Language Model

Classifier		MIT–Dev		NIST–Core	
		Correct	Accuracy	Correct	Accuracy
A	Phoneme T	63.6%	59.8%	62.0%	58.2%
B	Phoneme T+M	64.5%	61.1%	63.2%	60.1%
C	MFC Vowel/Consonant T	63.5%	59.6%	62.4%	58.5%
D	MFC Vowel/Cons T+M	64.9%	61.0%	63.1%	59.4%
E	Formant–Vowel/ Cons T	65.5%	60.6%	62.5%	57.1%
F	Formant–Vowel/ Cons T+M	65.5 %	61.1%	63.2%	58.4%
G	Fmt/MFCC–Vowel/ Cons T+M	66.8 %	62.0%	64.5%	59.7%
H	Glass et al. [29]				61.8%

As can be seen from Table 7.3, the phonetic recognition results are slightly worse at this point than the state–of–the–art performance (H). Overall, the performance on the NIST-Core test set is lower by about 1.5 to 2 percentage points. This is a well known fact resulting from the lack of vocabulary overlap of NIST–Core with the training set, NIST–Train, whereas MIT–Dev does have a considerable overlap with NIST–Train in terms of orthographic transcriptions. This indicates that today's phonetic recognition systems (tested on TIMIT) are not yet vocabulary–independent. We suspect that the implicit estimation of the apriori class probabilities (equivalent to using a unigram language model in

the phonetic search) is not estimated accurately enough by our MLP classifiers, especially in the multi–classifier case.

The performance of the multi–classifier experiments using cepstral features (C & D) are virtually identical to the experiments using a single classifier (A & B). This is not surprising considering that all three classifiers are using the same set of classification features. There are two main effects that could affect the recognition performance in a system using multiple classifiers. On one hand, more accurate classifiers could be built by separating the acoustically distinct consonants from the vowels and semi–vowels, and hence enabling the classifiers to create more powerful representations in the hidden layers. On the other hand, recognition error might be created due to the explicit selection of one of the classifiers by the lattice construction algorithm (tagging of the segment hypotheses with a broad phonetic identity).

Additionally, it should be noted that the results using the formant–based vowel/semi–vowel classifiers (E, F, & G) are comparable to the results using only cepstral–based classifiers (C & D), largely explainable by the almost identical classification performances (on hand–labeled test data). This is not self–evident however since the formant–based classifiers use additional features such as formant amplitude and pitch which cannot be computed as robustly as in the case of known segmentations. We have found that averaging the amplitude, pitch, and bandwidth features over the middle 80% of the segments increases the performance on machine–generated segments and hence also the phonetic recognition accuracy.

7.4 Summary

The phonetic recognition experiments, using context–independent phoneme models, described in this chapter demonstrate the validity of our segment–based approach. We discussed issues related to the probabilistic framework and noted that the estimation of $p(S)$, the probability of a segmentation is largely an unsolved problem. We proposed to use the broad category information of the acoustic segments used in the construction of the segment lattice to calculate boundary probabilities $p(b_k)$. As Glass et al. [30] have

pointed out, our approach (along with most others described in Chapter 2) is not considering the probability of all the competing hypotheses when computing the best–scoring path through the segment lattice. The issue of a proper probabilistic framework, taking into account the inaccuracies of the probability estimations by the segment classifiers, clearly deserves more attention in the future.

The context–independent phonetic–recognition results for unigram language models reported in this chapter are slightly below the state–of–the–art performance (under similar conditions). Nevertheless, these results are very encouraging and give credit to the idea of incorporating explicit speech knowledge into the recognition process.

Chapter 8

Conclusions and Future Work

The goal of this thesis was to create a flexible recognition framework for the explicit incorporation of speech knowledge into speech recognition algorithms. Below we summarize the major results from our research that demonstrate the partial accomplishment of this ambitious goal. This work should of course not be viewed as the definite answer to the question of the integration of speech knowledge into the recognition process, but rather as a step towards the goal of improving the accuracy and robustness of future speech recognition systems to a level that will allow for the technology to be used ubiquitously in everyday life. Section 8.2 contains an outline of future work we hope that will extend this body of work accordingly.

8.1 Thesis Summary and Contributions

The segment–based framework described in Chapter 3 provides a useful basis for integrating knowledge into speech–recognition systems. By using broad–phonetic classification scores as the basis for merging decisions, we were able to construct a segment lattice that contained on the average 6.0 segments (Table 3.3) for every segment of the hand–labeled transcription. The best aligned path through the lattice (using a dynamic programming search technique) has 2.5% insertions and 1.6% deletions with respect to the TIMIT transcription. We take advantage of the fact that each segment hypothesis in the lattice has a broad phonetic identity by inserting additional voiced stop segment hypotheses into the lattice structure in places where a closure–vowel transition was detected. After applying additional correction rules to the segment lattice, the alignment error was reduced to 1.2%

insertions and 1.1% deletions with an overgeneration of 7.00 segments per hand–labeled TIMIT segment.

Next, we trained a set of segment classifiers using static MFC–based features (Section 4.2). Three different classifiers were trained:

- a phonetic segment classifier with 40 phonetic categories performing at 74.0% correct classification on the MIT development test set,
- a vowel/semi–vowel classifier for 14 vowel and 4 semi–vowel categories with a classification accuracy of 71.6%, and
- a consonant classifier with the remaining 22 consonant categories which performs 78.8% correct classification.

We noted that there exists a mismatch between the development conditions of the classifiers and the usage thereof. The segment classifiers are trained and tested on hand–labeled segmentations, and used in a recognizer that does not have access to those perfect segmentations. When we evaluated the performance of the classifiers on machine–generated segmentations (using a search algorithm to find the best alignment of the segment lattice with the hand–labeled transcription) we observed a significant drop in classification accuracy, e.g. from 74.1% to 70.4% for the phonetic segment classifier (Section 4.3). Therefore, we added training examples based on machine–generated segmentations. The classifiers trained in this manner showed an improvement in accuracy on both the TIMIT and machine–segmented test vectors, e.g. the phonetic classifiers improved to 75.1% on TIMIT–segmented and to 72.8% on the machine–segmented ones. This demonstrates a general method for increasing the amount of training data (which generally results in better classification performance) while at the same time reducing the mismatch between training and usage conditions.

The second goal of this thesis was to demonstrate the potential of explicit formant features for the purpose of vowel classification. To that end, we developed a robust, N–best formant tracker which is described in Chapter 5. The tracking algorithm is based on the following assumptions:

- execute tracking only in sonorant regions,
- use a wild–card mechanism to compensate for errors in previous processing steps,
- find up to N "consistent" interpretations of the formant information, and
- delay the selection of the correct formant interpretation until after the classification and search.

In Chapter 6 we described vowel and semi–vowel classification experiments using these formant interpretations. We experimented with two methods for approximating the formant trajectories and found that using third–order Legendre polynomials outperformed (in terms of classification accuracy) the piecewise–linear regression method. Together with formant amplitude, formant bandwidth, pitch, and segment durations the Legendre coefficients and prediction error form a feature representation that classified the vowels and semi–vowels of the MIT development test set with virtually the same accuracy as the MFCC–based vowel/semi–vowel classifier (71.8% for the formant features versus 71.6% for cepstral features) for $N = 1$. Increasing N to 5 did not result in an improvement in classification accuracy. We discussed possible causes for this negative example in Section 6.4. In an additional "Oracle" experiment we demonstrated the potential for the N–best approach: by ignoring the magnitude of the posterior probabilities we achieved a 76.1% correct classification. Further research will have to investigate the possibility of either scaling the probability estimates or improvements to the pruning mechanism of the formant tracking algorithm to harness some of this potential for improvements in classification and recognition accuracies. In a final vowel classification experiment, we combined the above formant features with the cepstral–based features (for a total of 141 input features) and achieved the highest vowel/semi–vowel classification accuracy of 74.1%, when adding training vectors based on machine–generated segmentations.

The phonetic recognition experiments reported in Chapter 7 further demonstrate the validity of our segment–based framework with explicit formant features. Recognition accuracies of 62.0% on MIT–Dev and 60.1% on NIST–Core for unigram searches are

slightly worse than the best reported results under similar conditions (using context–independent models with unigram and bigram language models respectively). The results using vowel/semi–vowel classifiers based on explicit formant features showed a slight improvement in performance, although it is still below the performance of the state–of–the–art system. Nevertheless, these results are very encouraging and give credit to the idea of incorporating explicit speech knowledge into the recognition process. The chapter includes a discussion of search issues pertaining to segment–based systems.

The contributions of this thesis research are summarized in Table 8.1.

Table 8.1: Thesis Contributions

Thesis Research Contributions

- Creation of a segment–based recognition framework with broad category information,
- Demonstration of the benefits of including training vectors based on machine–generated segmentations,
- Creation of a robust, N–best formant tracker using a wild card mechanism,
- Demonstration of the potential of the N–best classification paradigm,
- Construction of vowel/semi–vowel classifiers based on explicit formant features,
- Validation of the segment–based recognition framework using multiple, specialized segment classifiers, and knowledge–based features, through phonetic recognition experiments.

Next, we describe some of the many possible extensions of this thesis research, in the hope that the reader will find inspirations for his or her own research.

8.2 Future Work

This research is meant to lay the foundations for a line of research exploring the use of explicit speech knowledge in a segment–based speech recognition paradigm. Our initial

focus was on using explicit formant features for vowel classification. We elected to work with the TIMIT database as an accepted standard to develop the basic technology. However, we realize that the true test for this technology lies in the performance under more difficult conditions, especially for telephone–quality speech. Therefore, in the first part of this section on future work, we will discuss anticipated research challenges and possible solutions to adapting the proposed technology to telephone–quality speech. In the second part, we outline other possible extensions to this basic paradigm.

8.2.1 Porting to Telephone–Quality Speech

There are several steps that have to be taken when porting our system, trained on the TIMIT database, to an acoustically distinct environment, listed in Table 8.2. Note that steps 3 and 4 are only necessary if formant–based features are used.

Table 8.2: Steps necessary to port the system from one environment to another.

Porting Steps
1. redefine the acoustic segmentation function,
2. retrain the acoustic–segment classifier,
3. adjust procedure to find formant candidates,
4. find pitch tracker for new environment, and
5. retrain segment classifier(s)

Research Challenges

From the list in Table 8.2 it can be seen that the biggest research challenges for porting SWISS to telephone–quality speech are the construction of an adequate pitch tracker and the reliable detection of formant candidates (steps 3 and 4). Contrary to popular belief, reliable and accurate pitch tracking is not a solved problem for telephone speech. Additionally, initial experiments with telephone speech show unreliable detection of formant candidates, especially for F3, using the peak–picking method described in this thesis. In

telephone speech, F3 can either vanish from the spectrum due to the bandwidth limitation (F3 > 3400 Hz) or channel distortions that tend to affect the weakest spectral peaks (i.e. F3) the most.

Possible Solutions

Recently, Welling and Ney proposed a formant estimation method based on digital resonators [89]. The dynamic programming algorithm reliably finds exactly k formant candidates per frame (by placing k resonators along the frequency axis). The drawback of this method is that there is not direct estimation of the correct number of formants present in the spectrum. The prediction error is always smaller of higher values of k. Initial work with this resonator method showed that in the cases where the number of formants present in the spectrum matches the number of resonators, the estimates are reliable. In the mismatched cases the following was observed (M is the number of formants in the spectrum): if $M = 3$ and $k = 2$ then the higher formant estimate tends to be placed between the true F2 and F3 unless F2 is considerably more prominent than F3. In the reverse situation, $M = 2$ and $k = 3$, the first formant tends to be approximated by two resonators. We expect that by combining the candidates found for $k = 2$ and for $k = 3$, resulting in 3–5 distinct formant candidates per frame, and using our N–best tracking algorithm, we should be able to overcome the assignment problems and produce a reliable formant tracker for telephone–quality speech. Additionally, some perceptual data suggests that measurements of F1 and F2 alone are able to separate the majority of the vowel categories (with the exception of the retroflexed vowels), leading to the speculation that simply using formant features based on F1 and F2 might prove powerful enough for the basic vowel classification.

8.2.2 Extensions to the Basic Paradigm

The following list of possible extensions to the basic paradigm are in no particular order.

Context–Dependent and Gender–Specific Modeling

The most obvious extension of the research presented in this thesis is the creation of context–dependent models. It has been observed numerous times that the accuracy of phonetic recognition improves when going from context–independent to context–dependent models. In the context of this research, context–dependent modelling could be achieved the "traditional" way [50] by creating triphonic models. Alternatively, the broad phonetic labeling of the segment lattice can be used to create (intermediate) broad–phonetic triphones (e.g. [closure]–/b/–[vowel]). In that case, it might be necessary to refine the choice of broad–phonetic categories by, for instance, splitting the [vowel] category into [front], [mid], and [back] vowels, to provide more detailed contextual information (especially with respect to place of articulation). Using broad–phonetic contexts will also limit the number of recognition units (compared to full phonetic triphone models). At the same time, using precomputed broad phonetic context will simplify the search algorithm because the application of segment and language model scores does not have to be delayed until after evaluating the next segment.

After adding the pitch feature to the set of classification features, we were unable to achieve further improvement using gender–specific classifiers and hence we did not pursue this issue further at this point. In light of potential difficulties of extracting reliable pitch information in different acoustic environments, we might want to reconsider using gender–specific models.

Biphone Models

A second suggested modification of the recognition inventory (along with the context–dependent models described above) is designed to address a typical segmentation problem: we have observed that certain boundaries, especially semi–vowel/vowel boundaries, are difficult to detect and place consistently. We therefore propose a data–driven biphone

creation algorithm. Such an algorithm would analyze segmentation and misclassification errors in context (e.g. frequent deletions of /w/ in front of back vowels) and then propose a set of biphones (e.g. /w/–/aa/ $\longrightarrow$ /w–aa/) to be added to the recognition inventory. In this case, biphones span two phonetic segments, in contrast to phonemes in left and/or right context which are sometimes referred to as biphones in the literature.

Consonant Classifier using Formant Features

In this thesis, we have concentrated our research on using formant features for vowel classification. As we have pointed out in the introduction, formants also play an important part in consonant identification. For example, while it is difficult to distinguish /n/ from /m/ by comparing the spectrum within the segment (nasal murmur), we can easily determine the nature of the nasal sound by detecting the place of articulation (via formant locations) in the adjacent segment(s): for /m/ F2 generally drops to about 1200Hz, whereas in the case of /n/, F2 moves closer to 1600Hz. Additionally, Weinstein et al. [88] use features based on spectrogram reading experience to classify consonant segments (nasals, fricatives, and stops). In particular, for their stop classification, they use a detailed analysis of the stop burst along with voicing information in the silence part preceding the burst, and formant transitions (F2 and F3) out of and into adjacent vowels as their classification features.

Relative Durations as Classification Features

It has been observed in the literature that the speaking rate is an important source of recognition errors. Normalization schemes for speaking rate in form of post–processing algorithms have been proposed [37]. Having access to the structure of the segment lattice would allow for the direct incorporation of relative duration information into the classification process.

Training on Errors

As suggested in Section 4.4, training the segment classifiers on false positives should result in a better phonetic recognition performance. During the training of the classifiers, we

generally present only positive examples of a category (along with training examples for all the other categories which indirectly act as negative examples). However, the classifier has never seen hypothesized segments that span more than one phonetic segment during training and hence we rely on the generalization capabilities of the classifier to assign low probabilities to all outputs. Training on errors is meant to remedy this by explicitly presenting the classifier with training examples of false positive along with enhanced features that will make it easier for the classifier to deduce that the current segment hypothesis is spanning more than one phonetic segment. Such features can be extracted either from the segment lattice structure or the classification scores of the acoustic segments. The difficulty will be to balance the training examples with positive and negative examples while maintaining low within–category variance and high inter–category variances.

Robustness to Noise

As indicated in the introduction to this thesis, we expect formant features to be more robust to ambient noise than cepstral features due to their focus on spectral peaks and the incorporation of global constraints. To realize this potential, we need to examine the detailed behaviour of the formant tracker in noisy conditions. Given the contextual information provided by the physical nature of formants, it should be relatively straightforward to develop algorithms to improve noise robustness.

Improving the Performance in the N–best Paradigm

The N–best experiments in Section 6.4 showed only modest improvements for $N = 3$ as compared to $N = 1$. We concluded that one of the reason is the lack of an efficient pruning mechanism for the list of interpretations produced by the current implementation of the formant tracker. Another is the problem of incorrect output levels of the MLP which prevents correct classifications from "winning" because incorrect classifications for other interpretations produce a higher output level. Rescaling of the outputs to better mimic the true probability densities is a major problem that requires fundamental advances in the study of neural networks. In the meantime, a combination of training on error procedures

might improve the N–best performance. Such schemes as introducing "Not–Phoneme–/xx/" units into the recognition vocabulary might help in suppressing high output levels for incorrect formant interpretations.

Segmentation of Sonorants based on Formant Information

Weinstein et al. [88] described a two–stage method for segmenting the speech: the first stage segments the speech into one of four broad acoustic categories: vowel–like, volume dip within vowel–like sound, fricative–like, and stops. The information of the formant tracking are then used to further segment the vowel–like sound regions. We envision a similar process based on our own formant tracking algorithm, and expect that this may help with the continued difficulties we have in segmenting between adjacent sonorant phonemes.

8.3 Famous Last Words

The above list of possible extensions to the research described in this thesis is by no means intended to be complete. We hope that the reader has found enough inspiration in this work to suggest further additions to the list. We intend this to be the beginning rather than the end in our quest to improve the accuracy of today's speech recognition systems.

The two quotes below have provided the guidance for this research:

"Garbage in – garbage out!"

"No risk – no gain!"

Bibliography

[1] Allen, J. How Do Humans Process and Recognize Speech? *IEEE Transactions on Speech and Audio Processing, vol. 2*, 4 (1994), 567–577.

[2] Atal, B., and Hanauer, S. Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. *Journal of the Acoustical Society of America, vol. 50*, 2 (1971), 637–655.

[3] Austin, S., Zavaliagkos, G., Makhou, J., and Schwartz, R. Speech Recognition using Segmental Neural Nets. In *Proceedings of ICASSP* (1992), pp. 625–628.

[4] Barnard, E. Optimization for Training Neural Net. *IEEE Transactions on Neural Network, vol. 3*, 2 (1992), 232–240.

[5] Barnard, E., and Cole, R. A. A Neural-Net Training Program based on Conjugate-Gradient Optimization. Tech. Rep. CSE 89-014, Oregon Graduate Center, 1989.

[6] Bishop, C. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[7] Bladon, A., Hendon, C., and Pickering, J. Towards an Auditory Theory of Speaker Normalization . *Language Communication, vol. 4* (1984), 59–69.

[8] Brugnara, F., DeMori, R., Giuliani, D., and Omologo, M. Improved Connected Digit Recognition using Spectral Variation Functions. In *Proceedings of ICSLP* (1992), pp. 627–630.

[9] Bush, M., and Kopec, G. E. Network–Based Connected Digit Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 35*, 10 (1987), 1401–1413.

[10] Calvert, D. *Descriptive Phonetics.* Thieme Inc., 1986.

[11] Carlson, R., and Glass, J. Vowel Classification Based on Analysis–by–Synthesis. In *Proceedings of ICSLP* (1992), pp. 575–578.

[12] Chen, S., and Wang, Y. Vector Quantization of Pitch Information in Mandarin Speech. *IEEE Transactions on Communications, vol. 38*, 9 (1990), 1317–1320.

[13] Chigier, B., and Brennan, R. Broad Class Network Generation Using a Combination of Rules and Statistics for Speaker Independent Continuous Speech. In *Proceedings of ICASSP* (1988), pp. 449–452.

[14] Chigier, B., and Leung, H. The Effects of Signal Representations, Phonetic Classification Techniques, and the Telephone Network. In *Proceedings of ICSLP* (1992), pp. 97–100.

[15] Cole, R., and Muthusamy, Y. Perceptual Studies on Vowels Excised from Continuous Speech. In *Proceedings of ICSLP* (1992), pp. 1091–1094.

[16] Cole, R., Roginski, K., and Fanty, M. English Alphabet Recognition with Telephone Speech. In *Proceedings of Eurospeech* (1991), pp. 342–345.

[17] Cole, R., Rudnicky, A., Zue, V., and Reddy, D. Speech as Patterns on Paper. In *Perception and Production of Fluent Speech,*, R. A. Cole, Ed. Lawrence Erlbaum Assoc., 1980, pp. 3–50.

[18] Cole, R., Stern, R., and Lasry, M. Performing Fine Phonetic Distinctions: Templates versus Features. In *Variability and Invariance in Speech Processes*, J. Perkel and D. Klatt, Eds. Lawrence Erlbaum Assocs., 1986, pp. 325–359.

[19] Cole, R., Yan, Y., Mak, B., Fanty, M., and T.Bailey. The Contribution of Consonants versus Vowels to Word Recognition in Fluent Speech. In *Proceedings of ICASSP* (1996), pp. 853–856.

[20] Cole, R., and Zue, V. Speech as Eyes See it. In *Attention and Performance VIII*, R. Nickerson, Ed. Lawrence Erlbaum Assoc., Hillsdale, NJ, 1980, pp. 475–494.

[21] Digalakis, V. *Segment–based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition.* PhD thesis, Boston University, Boston, MA, 1992.

[22] Digalakis, V., Ostendorf, M., and Rohlicek, J. Fast Algorithms for Phone Classification and Recognition Using Segment–Based Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 40*, 12 (1992), 2885–2896.

[23] Digalakis, V., Rohlicek, J., and Ostendorf, M. ML Estimation of a Stochastic Linear System with the EM Algorithm and its Application to Speech Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 1*, 4 (1993), 431–442.

[24] Flammia, G., Dalsgaard, P., Andersen, O., and Lindberg, B. Segment based Variable Frame Rate Speech Analysis and Recognition using a Spectral Variation Function. In *Proceedings of ICSLP* (1992), pp. 983–986.

[25] Forney Jr, G. D. The Viterbi Algorithm. *Proceedings of the IEEE, vol. 61* (1973), 268–277.

[26] Fukunaga, K. *Introduction to Statistical Pattern Recognition.* Academic Press, 1990.

[27] Gish, H., and Ng, K. A Segmental Speech Model with Applications to Word Spotting. In *Proceedings of ICASSP* (1993), pp. 447–450.

[28] Glass, J. *Finding Acoustic Regularities in Speech: Applications to Phonetic Recognition.* PhD thesis, Massachusetts Institute of Technology, 1988.

[29] Glass, J. Personal Communications, MIT, Cambridge, MA, 1996.

[30] Glass, J., Chang, J., and McCandless, M. A Probabilistic Framework for Feature–Based Speech Recognition. In *Proceedings of ICSLP* (1996), pp. 2277–2280.

[31] Goldenthal, W. *Statistical Trajectory Models for Phonetic Recognition.* PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1994.

[32] Hampshire, J., and Pearlmutter, B. Equivalence Proofs for Multi–Layer Perceptron Classifiers and the Bayesian Discriminant Function. In *Proceedings of the 1990 Connectionist Models Summer School* (San Mateo, CA, 1990), D. Touretzky, J. Elman, T. Seijnowski, and G. Hinton, Ed., Morgan Kaufmann, pp. 157–172.

[33] Hermanski, H. Perceptual Linear Predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America, vol. 87*, 4 (1990), 1738–1752.

[34] Hermansky, H., and Cox, L. Perceptual Linear Predictive (PLP) Analysis–Resynthesis Technique. In *Proceedings of Eurospeech* (1991), pp. 415–418.

[35] Hetherington, I. L., Phillips, M., Glass, J., and Zue, V. A* Word Network Search for Continuous Speech Recognition. In *Proceedings of Eurospeech* (1993), pp. 1553–1536.

[36] Hu, Z., J.Schalkwyk, Barnard, E., and Cole, R. Speech Recognition using Syllable–like Units. In *Proceedings of ICSLP* (1996), pp. 1117–1120.

[37] Jones, M., and Woodland, P. C. Using Relative Duration in Large Vocabulary Speech Recognition. In *Proceedings of Eurospeech* (1993), pp. 311–314.

[38] Kass, M., Witkin, A., and Terzopoulos, D. Snakes: Active Contour Models. *International Journal of Computer Vision* (1987), 321–331.

[39] Klatt, D. Review of ARPA Speech Understanding Project. *Journal of the Acoustical Society of America, vol. 62*, suppl. 1 (1977), 1345–1366.

[40] Klatt, D. Prediction of Perceived Phonetic Distance from Critical–Band Spectra: a First Step. In *Proceedings of ICASSP* (1982), pp. 1278–1281.

[41] Kopec, G. A Family of Formant Trackers Based on Hidden Markov Models. In *Proceedings of ICASSP* (1986), IEEE, pp. 1225–1228.

[42] Kopec, G. Formant Tracking Using Hidden Markov Models and Vector Quantization. *IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 34*, 4 (1986), 709–729.

[43] Kreidler, C. *The Pronunciation of English: a Course Book in Phonology.* Basil Blackwell Inc., 1989.

[44] Krishnan, S., and Rao, P. V. S. Segmental Phoneme Recognition using Piecewise Linear Regression. In *Proceedings of ICASSP* (1994), pp. 49–52.

[45] Ladefoged, P. *A Course in Phonetics*, third edition ed. Harcourt Brace Jovanovich College Publishers, 1993.

[46] Lamel, L., R.Kassel, and Seneff, S. Speech Database Development: Design and Analysis of the Acoustic–Phonetic Corpus. In *Proc. DARPA Speech Recognition Workshop* (1986), pp. 100–109.

[47] Laprie, Y. Optimum Spectral Peak Teack Interpretation in Terms of Formants. In *Proceedings of ICSLP* (1990), pp. 1261–1264.

[48] Laprie, Y. A New Paradigm for Reliable Automatic Formant Tracking. In *Proceedings of ICASSP* (1992), pp. 201–204.

[49] Laprie, Y., and Berger, M. A New Paradigm for Reliable Automatic Formant Tracking. In *Proceedings of ICASSP* (1994), pp. 201–204.

[50] Lee, K. *Large Vocabulary Speaker–Independent Continuous Speech Recognition: The SPHINX System.* PhD thesis, Computer Science Department, Carnegie Mellon University, 1988.

[51] LEUNG, H. *The Use of Artificial Neural Networks for Phonetic Recognition.* PhD thesis, Dept. of Electrical Engineering, Massachusetts Institute of Technology, 1989.

[52] LEUNG, H., CHIGIER, B., AND GLASS, J. A Comparative Study of Signal Representations and Classification Techniques for Speech Recognition. In *Proceedings of ICASSP* (1994), pp. 680–683.

[53] LEUNG, H., GLASS, J., PHILLIPS, M., AND ZUE, V. Detection and Classification of Phonemes Using Context–Independent Error Back–Propagation. In *Proceedings of ICSLP* (1990), pp. 1061–1064.

[54] LEUNG, H., HETHERINGTON, I., AND ZUE, V. Speech Recognition Using Stochastic Explicit–Segment Modeling. In *Proceedings of Eurospeech* (1991), pp. 212–215.

[55] LEUNG, H., HETHERINGTON, I., AND ZUE, V. Speech Recognition using Stochastic Segment Neural Networks. In *Proceedings of ICASSP* (1992), pp. 613–616.

[56] LEVINSON, S. Speech Recognition Technology: a Critique. In *Voice Communication Between Humans and Machines* (Washington D.C., 1994), National Academy Press, pp. 159–164.

[57] LINDBLOM, B., AND STUDDERT-KENNEDY, M. On the Role of Formant Transitions in Vowel Recognition. *Journal of the Acoustical Society of America, vol. 42* (1967), 830–843.

[58] LIPPMANN, R. Speech Perception by Humans and Machines. In *Proceedings of the Workshop on the Auditory Basis of Speech Perception* (1996), W. Ainsworth and S. Greenberg, Eds., Keele University Press, pp. 309–316.

[59] LJOLJE, A. High Accuracy Phone Recognition using Context Clustering and Quasi–Triphonic Models. *Computer Speech and Language, vol. 8*, 1 (1994), 129–151.

[60] MAKHOUL, J. Linear Prediction: A Tutorial Review. *Proc. IEEE, vol. 63*, 2 (1975), 561–580.

[61] MCCANDLESS, S. An Algorithm for Automatic Formant Extraction Using Linear Prediction Spectra. *IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 22* (1974), 135–141.

[62] MENG, H. *The Use of Distinctive Features for Automatic Speech Recognition.* Master's thesis, Massachusetts Institute of Technology, 1991.

[63] Meng, H., and Zue, V. Signal Representation Comparison for Phonetic Classification. In *Proceedings of ICASSP* (1991), pp. 285–288.

[64] Mermelstein, P., and Davis, S. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 23*, vol. 1 (1975), 67–72.

[65] Miller, J. Auditory–Perceptual Interpretation of the Vowel. *Journal of the Acoustical Society of America, vol. 85*, 5 (1989), 2114–2134.

[66] Olive, J. Automatic Formant Tracking by a Newton–Raphson Technique. *Journal of the Acoustical Society of America, vol. 50* (1971), 661–670.

[67] Oppenheim, A., and Schafer, R. *Discrete–Time Signal Processing.* Prentice Hall, 1989.

[68] Ostendorf, M., and Roukos, S. A Stochastic Segment Model for Phoneme–Based Continuous Speech Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37*, 12 (1989), 1857–1869.

[69] Peterson, G., and Lehiste, I. Duration of Syllable Nuclei in English. *Journal of the Acoustical Society of America, vol. 30* (1960), 693–703.

[70] Phillips, M. Automatic Discovery of Acoustic Measurements of Acoustic Classification. *Journal of the Acoustical Society of America, vol. 84*, Suppl. 1 (1988), 216.

[71] Pinker, S. *The Language Instinct.* Morrow, New York, NY, 1994.

[72] Poritz, A. Hidden Markov Model: a Guided Tour. In *Proceedings of ICASSP* (88), pp. 7–13.

[73] Rabiner, L. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Readings in Speech Recognition*, A. Waibel and K. F. Lee, Eds. Morgan Kaufman Publishers, Inc., 1990, pp. 267–296.

[74] Rabiner, L., and Schafer, R. *Digital Signal Processing of Speech Signals.* Prentice–Hall, Inc., 1978.

[75] Robinson, T. Several Improvements to a Recurrent Error Propagation Phone Recognition System. Tech. Rep. *CUED/TINFENG/TR.82*, Cambridge University Engineering Dept., 1991.

[76] Rummelhart, D., and McClelland, J. *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, vol. 1. MIT Press, 1987.

[77] Schafer, R., and Rabiner, L. System for Automatic Formant Analysis of Voiced Speech. *Journal of the Acoustical Society of America, vol. 57*, 634–648 (1970).

[78] Schmid, P., and Barnard, E. Robust, N–best Formant Tracking. In *Proceedings of Eurospeech* (1995), pp. 737–740.

[79] Schmid, P., Cole, R., Fanty, M., Bourlard, H., and Haessen, M. Real-Time, Neural Network-Based, French Alphabet Recognition with Telephone Speech. In *Proceedings of Eurospeech* (1993), pp. 1723–1726.

[80] Seneff, S. A Joint Synchrony/Mean–Rate Model of Auditory Speech Processing. *Journal of Phonetics, vol. 16* (1988), 55–76.

[81] Stern, P., Eskenazi, M., and Memmi, D. An expert system for speech spectrogram reading. In *Proceedings of ICASSP* (1986), pp. 1193–1196.

[82] Strange, W. Evolving Theories of Vowel Perception. *Journal of the Acoustical Society of America, vol. 85*, 5 (1989), 2081–2087.

[83] Talkin, D. Speech Formant Trajectory Estimation Using Dynamic Programming with Modulated Transition Costs. *AT&T Internal Memo MH 11222 2924 2D-410*, AT&T, 1987.

[84] Talkin, D. *ESPS Manual.* Entropic Research Lab., Inc., 1993.

[85] Tebelskis, J., and Waibel, A. Performance Through Consistency: MS–TDNN's for Large Vocabulary Continuous Speech Recognition. In *Advances in Neural Information Processing 5* (1992), J. C. S. Hanson and L. Giles, Eds., Morgan Kaufman Publishers, Inc., pp. 696–701.

[86] van Vuuren, S. Pitch Detection. Tech. Rep. (to be published), Dept. of Computer Science and Engineering, Oregon Graduate Institute of Science and Technology, Portland, OR, 1996.

[87] Voice Information Associates. Automatic Speech Recognition: A Study of the World–Wide Market, 1995.

[88] Weinstein, C., McCandless, S., Modshein, L., and Zue, V. A System for Acoustic–Phonetic Analysis of Continuous Speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 23*, 1 (1975), 314–327.

[89] Welling, L., and Ney, H. A Model for Efficient Formant Estimation. In *Proceedings of ICASSP* (1996), pp. 797–800.

[90] Wolf, J., and Woods, W. The HWIM speech understanding system. In *Proceedings of ICASSP* (1977), pp. 784–787.

[91] Wooters, C., and Morgan, N. Acoustic Subword Models in the Berkeley Restaurant Project. In *Proceedings of ICSLP* (1992), pp. 1551–1554.

[92] Zue, V., and Cole, R. Experiments on spectrogram reading. In *Proceedings of ICASSP* (1979), pp. 116–119.

[93] Zue, V., Glass, J., Goodine, D., Leung, H., Phillips, M., Polifroni, J., and Seneff, S. Recent Progress on the SUMMIT System. In *Proceedings of the DARPA Speech and Natural Language Workshop* (1990), Morgan Kaufmann, Inc., pp. 1–11.

[94] Zue, V., Glass, J., and Seneff, S. The MIT SUMMIT Speech Recognition System: A Progress Report. In *Proceedings of the DARPA Speech and Natural Language Workshop* (1989), pp. 179–189.

[95] Zue, V., and Lamel, L. An Expert Spectrogram Reader: A Knowledge–Based Approach to Speech Recognition. In *Proceedings of ICASSP* (1986), pp. 1197–1200.

Appendix A

Distance Functions

A.1 Overview

As mentioned in Chapter 3, most segmentation methods rely on a distance measure $d(\vec{x}, \vec{y})$ to detect spectral changes; for instance:

$$d(\vec{x}, \vec{y}) = \quad (\vec{x} - \vec{y})^T A(\vec{x} - \vec{y}) \qquad \text{Squared Difference Function}$$

where $\vec{x}$ and $\vec{y}$ are feature vectors representing the spectrum to the left and the right of the potential boundary respectively.

It has been observed that this function is sensitive to changes in signal gain. Therefore, the Spectral Variation Function [8, 24], which measures the angle between the two feature vectors, has been proposed:

$$SVF \quad = \quad d(\vec{x}, \vec{y}) = \quad 1 - \frac{\vec{x}^T A \vec{y}}{|\vec{x}|\,|\vec{y}|} \qquad \text{Spectral Variation Function} \tag{A.1}$$

Both distance functions use a matrix A to define the distance metric to be used:

Identity Matrix	$\longrightarrow$	Euclidian Distance
Diagonal Matrix	$\longrightarrow$	Mahalanobis Distance
$A(\vec{z})$	$\longrightarrow$	Context–dependent Distance Function

(where the vector $\vec{z}$ represents the current acoustic environment).

A.2 Context–Dependent Spectral Variation Function

The estimation of the context-sensitive weighting coefficients $c_i = a_{ii}$ (the diagonal elements of the distance matrix A) is an ideal application for a Mulitlayer Perceptron (MLP). The input to the neural network is the context vector $\vec{z}$ and the outputs are the coefficients c_i as shown in Figure (A.1). The nodes in the input and the output layer are linear whereas the nodes of the hidden layer use a sigmoidal activation function.

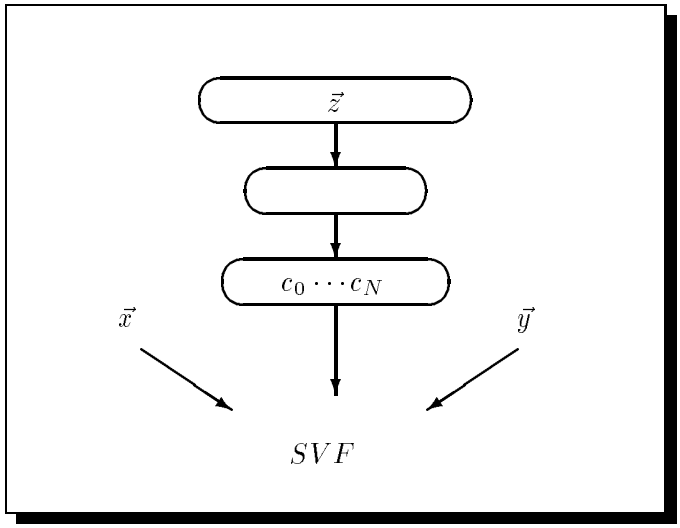

Figure A.1: MLP for computing the context-sensitive weighting coefficients

In order to perform the weight updating we need to compute the gradient of the total error with respect to the coefficients c_i. Unfortunately, there are no target values of the c_i's for computing the total error. However, for any given $\vec{x}$, $\vec{y}$ and $\vec{z}$ we know what the desired value of SVF is namely:

$$SVF(\vec{x}, \vec{y}, \vec{z}) = \begin{cases} 1 & \text{there is a phonetic} \\ & \text{boundary between } \vec{x} \text{ and } \vec{y} \\ 0 & otherwise \quad . \end{cases}$$

Using the chain rule, we can compute the gradient for the output layer nodes as follows:

$$\frac{\partial(MSE)}{\partial c_i} = \frac{\partial(MSE)}{\partial(SVF)} \cdot \frac{\partial(SVF)}{\partial c_i} = (O - T) \cdot \frac{\partial(SVF)}{\partial c_i} \quad , \tag{A.2}$$

where T is the desired function value of SVF, O the actual function value of SVF, and MSE the back-propagated error to the output layer of the MLP.

Using the following definitions:

$$S_x = \sum_{i=0}^{N} c_i^2 x_i^2 \quad , \quad S_y = \sum_{i=0}^{N} c_i^2 y_i^2 \quad , \quad S_{xy} = \sum_{i=0}^{N} c_i^2 x_i y_i \quad , \tag{A.3}$$

we can rewrite the equation (A.1) as:

$$SVF = \frac{S_{xy}}{(S_x \cdot S_y)^{\frac{1}{2}}} \quad . \tag{A.4}$$

Computing the partial derivative then becomes:

$$SVF' = \frac{\partial SVF}{\partial c_i} = \frac{S'_{xy}\left[(S_x S_y)^{\frac{1}{2}}\right] - S_{xy}\left[(S_x S_y)^{\frac{1}{2}}\right]'}{S_x \cdot S_y} \quad . \tag{A.5}$$

We can further simplify equation (A.5) to:

$$SVF' = -\frac{1}{2} \cdot \frac{c_i\left(2S_x S_y x_i y_i - S_{xy}\left[S_x y_i^2 + S_y x_i^2\right]\right)}{(S_x \cdot S_y)^{\frac{3}{2}}} \quad . \tag{A.6}$$

Figure A.2 shows an example of the context–dependent SVF.

A.3 Discussion

Initial experiments with the context–dependent SVF segmentation function showed a mixed picture. On one hand, the scaling mechanism was able to approximate the desired boundary probabilities (1.0 for the boundary frames and 0.0 elsewhere) more closely than SVF alone, and hence can potentially be beneficial in a probabilistic framework that estimates segment probabilities $p(s_i)$. On the other hand, in cases where the "generating function", in this case the SVF, did not detect any significant spectral change, the context–dependent extension was not able to rectify the situation and generate the correct boundary probabilities. As a matter of fact, doing so would require the classifier

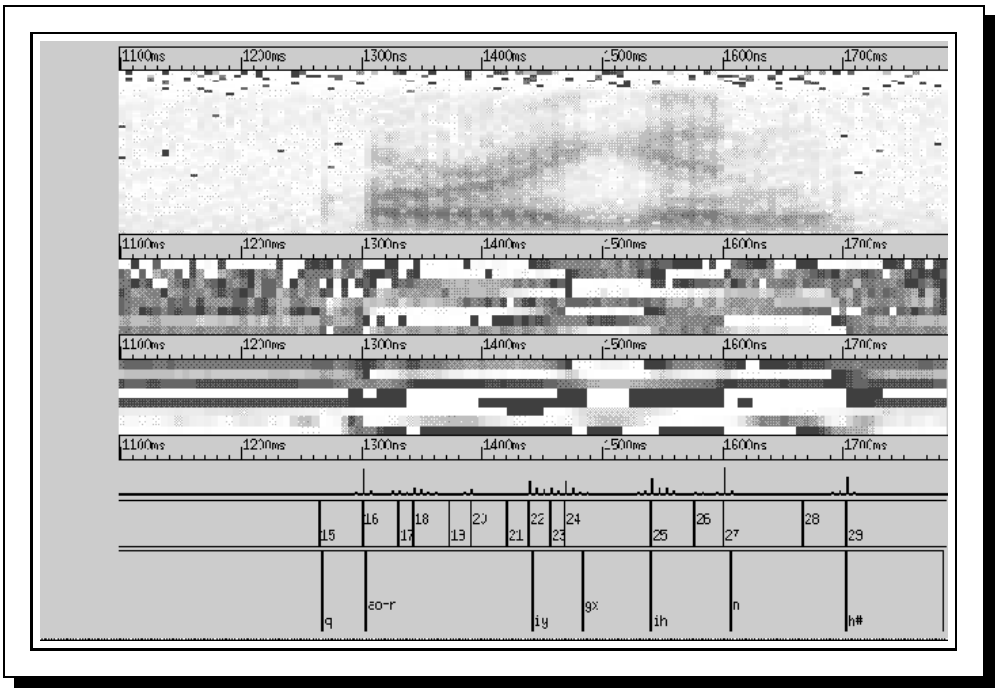

Figure A.2: Example of a context–dependent SVF screen dump. The figure shows from top to bottom: speech waveform, spectrogram, PLP coefficients, outputs of the SVF network (the scaling values for the distance function, CD–SVF function values, initial segmentation based on the above function values, and the TIMIT hand labels.

to basically detect the presence of a boundary and generate very large scaling terms to compensate for the missing boundary indications from the SVF. This is equivalent in complexity to the original task of detecting phonetic boundaries.

However, we nevertheless believe in the potential of the basic idea (estimating parameters of a distance function from data). Future research involving other "generating function" will show if our optimism is justified.

Biographical Note

Philipp Heinz Schmid was born on September 4, 1964 in Bern, Switzerland. After graduating from secondary school in Vechigen in 1979, he went on to attend the Realgymnasium Neufeld in Bern. In 1982/83 he participated in a student exchange program organized by American Field Service in which he spent a year at Wilson High School in Portland, Oregon. A year later, he graduated from the Realgymnasium and entered academia by enrolling as a student in computer science (informatics) at the University of Bern. During his first 3 years, he interspersed studying with service in the Swiss Army, where he currently holds the rank of Lieutenant.

The highlight of his undergraduate academic career was the presentation of a research paper at the International Neural Network Conference in Paris in 1990. He graduated from the Department of Informatics and Applied Mathematics in December 1990 with a thesis on a two–stage learning algorithm for neural networks.

Encouraged by the previous positive experience of living in America, he decided to return to America to pursue a Ph.D. degree in computer science. Prof. Ron Cole convinced him that Oregon Graduate Institute was a special place to work. He started studying for the qualifying exams at OGI in the summer of 1991, which he passed in the fall of 1991 without having attended any preparatory classes. The following summer, he won the first prize for the best student research paper and presentation.

Initial research into automatic pronunciation learning convinced him that the future of speech recognition lay in segment–based speech recognition systems. This led to a series of research projects that culminated in his Ph.D. thesis in October 1996. In the process, he published in and attended several national and international conferences, collaborated with international companies, and travelled throughout the United States and the world.