Automatic Language Identification with Sequences of Language-Independent Phoneme Clusters

Kay Margarethe Berkling
B.S. Computer Engineering, Syracuse University , 1991
B.S. Mathematics, Syracuse University , 1991
B.A. French, Syracuse University , 1991
B.A. German, Syracuse University , 1991

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

© Copyright 1996 by Kay Margarethe Berkling
All Rights Reserved

The dissertation " Automatic Language Identification with Sequences of Language-Independent Phoneme Clusters " by Kay Margarethe Berkling has been examined and approved by the following Examination Committee:

Etienne Barnard
Associate Professor
Thesis Research Adviser

Ronald A. Cole
Professor

Todd Leen
Associate Professor

Marc A. Zissman
Staff, Speech Systems ology Group
Lincoln Laboratory
Massachusetts Institu Technology

Dedication

to my parents who gave me the wings to fly

Acknowledgements

I want to thank my advisor, Dr. Etienne Barnard: he always knew what I was asking and what the answer was. I am indebted to Dr. Ronald Cole for teaching me to be self confident. Thanks to Dr. Marc Zissman for the helpful discussions and advice. Thanks to Dr. Todd Leen who always believed in me and called me "Dr. Berkling" a long time ago. Special thanks to my father for inciteful discussions about my work, Salma and Imran who inspired me to apply for graduate school, and of course my mother who always knew that I would get a Ph.D.. I am also very grateful to Dr. Kano, who got me started in Engineering and Yeshwant Muthusamy because he alone is responsible for getting me into the speech group. Yeshwant was always a good influence and a helpful and funny office mate. Thanks also go to the database labelers for making my research possible. And Jacques, thanks for teaching me about pointers in C and populating Cascade at night. For making the last couple of months easier I won't forget all the help I got from Hal Miller. Most importantly, I want to thank Roger Barga because he is the only reason that I finished this thesis. He taught me how to focus, was the chief editor of my thesis and always believed in me.

Contents

Dedication iv

Acknowledgements v

Abstract xiv

1 Introduction 1

1.1 From Speech Recognition to Language Identification 2

1.1.1 Acoustic-Phonetic Function 4

1.1.2 Alignment of Speech 5

1.1.3 Structural Feature Extraction 7

1.1.4 Language Identification 7

1.2 A New Approach 9

1.2.1 *Linguistic Motivation* 10

1.2.2 Multi-lingual Speech Representation 11

1.2.3 Language Identification 13

1.2.4 Contribution 14

2 Previous Work 16

2.1 Speech Representation 18

2.1.1 From Broad Category to Fine Phonemic Modeling 18

2.1.2 From Fine Phonemic Models to Clustered Phoneme Modeling 21

2.2 Structural Feature Extraction 23

2.2.1 Language Identification with Word/Sequence Spotting 24

2.2.2 Topic Identification with Word/Sequence Spotting 25

2.3 Related Methods 26

2.3.1 Language Identification without Phonemes 27

2.3.2 Other Related Work 28

3 Feature Modeling and Discrimination of Languages **30**
3.1 Speech Unit derivation 31
3.2 Feature Modeling 34
3.3 Feature Selection 38
3.4 The Complete Algorithm 43
3.5 Language Classification 46

4 Effect of Inaccurate Alignment on Language Discrimination **48**
4.1 Modeling True Data 49
4.1.1 Specifying the Model Language τ 49
4.1.2 Language Discrimination 50
4.2 Modeling Data with Misrecognitions 53
4.2.1 Modeling Independent Features 53
4.2.2 Modeling Correlated Features 56
4.2.3 Language Discrimination 60
4.3 Modeling Reconstructed True Data 62
4.3.1 Feature Modeling 63
4.3.2 Discrimination with Inexact Sequences 66
4.4 Classifying Model Language τ 68
4.4.1 Exact vs. Inexact Sequence Matching 68
4.4.2 Case 1: Adding Common Sequences 70
4.4.3 Case 2: Adding Neutral Sequences 72
4.4.4 Case 3: Adding Opposing Sequences 75

5 Language Identification of Telephone Speech **78**
5.1 The Data 79
5.1.1 Data Collection 80
5.1.2 Languages 80
5.1.3 Corpus Statistics 80
5.2 Speech Recognition - Alignment 85
5.2.1 Neural Network based Phoneme Classification 85
5.2.2 Aligning the Speech 86
5.3 Speech Representation 87
5.4 Language Identification System 89
5.4.1 System Design 90
5.4.2 Sequence Selection 91
5.4.3 Sequence Spotting 92
5.4.4 Language Identification 93

5.5 Results 96
5.5.1 Statistics-based Language Classification 96
5.5.2 Non-linear Language Classification 98
5.5.3 Evaluating the Impact of Alignment 104

6 Conclusion 107
6.1 Summary of Thesis 107
6.2 Present and Future of Language Identification 109
6.2.1 Design Issues for Language Identification Systems 109
6.2.2 This Thesis and the Future 111

Bibliography 115

A Labeling Conventions 124

B Label Statistics 125

C English vs. German 130
C.1 Clustering Trees 130
C.2 Merged Classes 131
C.3 Disallowed Merges 132
C.4 Word list to discriminate EN vs. GE 133

D Results using Neural Network Implementation 134

E Inexact Sequence Matching 135
E.1 Motivation 135
E.2 Terminology 138
E.3 Distance Function 139
E.4 Weighting Factor 143
E.5 Feature selection 146
E.6 Language Identification 147
E.7 Results 148

Biographical Note 149

List of Tables

3.1 Example of specifications for feature parameters. Assuming normal distributions, only the mean and standard deviation have to be specified. 47

4.1 Definitions 51
4.2 Confusion matrix due to alignment with independent recognition of labels. 54
4.3 Confusion matrix due to the alignment with interdependent recognition of labels. . 56
4.4 Specifications for model language τ which has one language discriminating feature. 61
4.5 Specification for case 1 71
4.6 Specification for case 2 72
4.7 Specifications for model language τ which has one language discriminating feature. 72
4.8 Specification for case 3 75
4.9 Specifications for model language τ which has one language discriminating feature. 75

5.1 Number of files exceeding various durations. 81
5.2 Table of labels for unmerged label set of 95 phonemes 81
5.3 Table of premerged labels within final set of 95 phonemes 82
5.4 Labels with word examples 83
5.5 Labels with word examples 84
5.6 Summary of Results from Alignment 90
5.7 Error rates when classification assumes a normal distribution. 97
5.8 Error rates using neural network with weighting of features 99
5.9 Error rates using neural network without weighting of features ($\alpha_i = 1$). 99
5.10 Comparative Results on standard NIST 1994 test set 101
5.11 Comparative Results on standard NIST 1996 test set 102
5.12 Simulating Better Alignment. 104
5.13 Average Accuracy of phoneme classes corresponding to different N. $N = 1.0$ corresponds to the original aligned data with 25% accuracy after alignment. . . . 105

6.1 General Evaluation Criteria of Language Identification Systems 110
6.2 Evaluation Criteria of Language Identification Systems (NIST 1996) 111

B.1 Phoneme Frequencies in Labeled files 125

B.2 Phoneme Frequencies in Labeled files 126
B.3 Phoneme Frequencies in Aligned files 127
B.4 Phoneme Frequencies in Aligned files 128
B.5 Phoneme Frequencies after clustering to 59 phonemes (aligned) 129

C.1 Table of merged classes 131
C.2 Disallowed merges for single phonemes. 132
C.3 Disallowed merges for phoneme classes. 132
C.4 The List of Words discriminating English vs. German. 133

D.1 Number of files classified from test set. 134
D.2 Names of misclassified files from test set. 134

List of Figures

1.1 Basic processing components of a general speech recognition system. 3
1.2 Sample wave file of the spoken word "Language" 4
1.3 Multi-lingual Speech Representation 6
1.4 The quality of structural features used by the application depends on the speech representation. 8
1.5 Modules of the LID System chosen for this thesis 11
1.6 Modules of the LID System chosen for this thesis: The system consists of a phoneme recognizer, followed by an automatic alignment of the speech with the recognized phonemes. Finally, features are derived based on the sequences discriminating German and English. 13

2.1 Information Sources Used in Multi-lingual Speech Recognition Systems 17
2.2 Language Identification Using Phoneme Recognition and Phonotactic Language Modeling 19
2.3 Language identification by spotting for mono- and ploly-phonemes. Each subsystem returns a likelihood score. The maximum score identifies the language of the speech. 22

3.1 Effect of confusion matrix on expected mutual information. **b** and **1-b** denote the prior probability of the two occurring classes. **a** is defined in Equation 3.2 32
3.2 Example clustering tree showing the sequence of merges from phonemes to phoneme clusters 33
3.3 Normal Distribution. 35
3.4 s2 as a function of time (in segments) for different values of u. 38
3.5 Combining two features linearly for optimal discrimination. 39
3.6 Derivation of α 41
3.7 Flowchart for estimating the error of an LID system and deriving the appropriate set of labels. 45
3.8 Actual error from model language τ and estimated errors from Bhattacharyya distance and Bayes' error. Plot shows fractional error as a function of time (measured in terms of the number of phonemic segments observed). 46

4.1 Channel with recognition probabilities. 50
4.2 Misrecognition probabilities due to alignment. 53
4.3 Classification error as a function of time (in terms of segments) for various values of $P(X|A) = p_{AX}$. 55
4.4 Discrimination error using exact sequence matching of x as a function of p_{ax} and p_{bx}. 62
4.5 Alignment process and reverse process characterictics. 63
4.6 Classifier based on reestimated true data. 66
4.7 Discrimination is better when using two separate features rather than reestimating the distribution of $\tilde{a}$ 70
4.8 Surface plot for the difference in error for exact vs. inexact sequence matching using specifications given in Table. 4.5. The surface would lie below zero if inexact matching were to outperform exact sequence matching. 71
4.9 Surface plot for the difference in error for exact vs. inexact sequence matching using specifications given in Eq. 4.34. The surface lies below zero when inexact matching outperforms exact sequence matching ($p_{ax} < p_{bx}$). 73
4.10 Discrimination is best when using the alternate feature y rather than x or reestimating the distribution of $\tilde{a}$ 73
4.11 Plot of discrimination error for case 2. Exact sequence matching outperforms inexact sequence matching when the correct label is matched. 74
4.12 Surface plot for the difference in error for exact vs. inexact sequence matching using specifications given in Eq. 4.8. The surface lies below zero when inexact matching outperforms exact sequence matching. 76
4.13 Plot of discrimination error for case 2. Treating features separately is preferable to combining features in inexact sequence matching. 77

5.1 Sampling intervals for the PLP features. The solid boxes indicate the frames for which PLP coefficients are computed. Dashed boxes indicate skipped frames. . . . 85
5.2 Automatic labeling of incoming utterance based on Viterbi search and neural network outputs. 86
5.3 Flowchart of the Language Identification System. 91
5.4 Estimated error as a function of time and number of sequences used as features . . 92
5.5 Plot of estimated error for each sequence corresponding to the sorted list. 93
5.6 Estimate and Actual Classification Error Probability as a Function of Number of Processed Speech Segments Using Statistical Approach. 94
5.7 Neural network based setup for language identification 95

5.8 Ratio of complete to simplified Bhattacharyya distance measure as a function of the number of sequences in the list. 97
5.9 Classification Error Probability as a Function of Time Using the Neural Network Approach. 98
5.10 Performance before clustering is inferior to performance after clustering. 100
5.11 % Correct classification for neural netword classifier as a function of the number of features used. Results shown before (95 classes) and after (59 classes) clustering for test set of feature vectors. 103
5.12 The error curve for identifying German and English as a funcition of the time (in ms), where each phoneme segment is artificially set to 10ms. 103
5.13 Effect of alignment on language identification error: for different alignment accuracies, the classification rate is shown as a function of the number of phonemes in the classified utterance. 106

6.1 Modules of the LID System chosen for this thesis 109
6.2 Modules of extended LID System The system consists of a phoneme recognizer, followed by an automatic alignment of the speech with the recognized phonemes. Structural features for each language are derived based on their discriminating sequences. 113

A.1 Table of Worldbet symbols. 124

C.1 Clustering of phonemes. The last shown merge represents the forbidden merge. . . 130

E.1 Example of speaker dependent pronunciations of same word. 136
E.2 Example of pronunciations overlapping languages. 137
E.3 Space of All Sequences. A,B, and C represent the centers of the three sets. Each set is associated with a radius shown by the line. Sets may overlap. 138
E.4 Plot of confidence for bigram probability as a function of $P(L|A)$. 142
E.5 Grouping sequences sorted by $P(L|A) = \beta$. 145
E.6 Comparing results for exact vs. inexact string matching. Using 50 features results are plotted for the test set of German and English. 148

Abstract

Automatic Language Identification with Sequences of Language-Independent Phoneme Clusters

Kay Margarethe Berkling, Ph.D.
Oregon Graduate Institute of Science & Technology, 1996

Supervising Professor: Etienne Barnard

Automatic Language Identification involves analyzing language-specific features in speech to determine the language of an utterance without regard to topic, speaker or length of speech. Although much progress has been made in recent years, language identification systems have not been built on detailed underlying theory or linguistically meaningful design criteria. This thesis is motivated by the belief that features used to discriminate between languages should be linguistically sound; the result is a unique combination of design, theory and implementation.

In this thesis a "word-spotting" algorithm is introduced motivated by a perceptual study [82] reporting that human subjects use language-dependent phonemes and short sequences to identify languages. In order to find an optimal set of phoneme-like tokens to represent speech in a linguistically meaningful way, a mathematical model of the discrimination between two languages is developed. This model permits the automatic design of a token representation of speech by selecting a list of discriminating "words" in a data-driven manner. The resulting system has the flexibility to automatically take into account the inherent structure of the languages to be discriminated. A second mathematical model is

developed to measure the impact of inaccurate automatic alignment of tokens on language discrimination. This model indicates why some algorithms aiming to compensate for these inaccuracies have not been successful. The theoretical models and the "word"-spotting algorithm have been implemented and validated on both generated and real-world speech data.

This dissertation makes several significant contributions: the design of a simple and linguistically sound language-identification module; a flexible automatic feature extraction algorithm; a mathematical model to estimate the discriminability of two languages; and a mathematical model to capture the impact of inaccurate alignment on the discriminability of two languages.

Chapter 1

Introduction

In this thesis we address the issue of complexity in automatic language identification systems. Automatic language identification (ALI) refers to a computer classifying the language of human speech input independent of speaker and topic. Classification is performed based on a set of features which are extracted from a tokenized representation of speech.

Token sets used for speech representation are usually phonemes which are defined by linguists to cover the complete inventory of speech sounds within a given language. Since phonemes are by definition language dependent, the difficulty of designing a linguistically sound language identification system lies in finding a token set which is valid across languages. Today's state-of-the-art language identification systems are typically constructed by using language dependent speech recognition systems as basic building blocks. Such a construction however can result in the high complexity due to the detailed modeling of phonemes from a potentially large number of languages. Reducing this complexity by modeling phonemes from a partial set of languages has in the past resulted in a speech representation with little linguistic meaning.

We introduce a new approach to automatic language identification, which does not combine language-dependent phoneme models but creates a single set of clustered phonemes valid across languages. We address the issue of complexity by creating a mathematical model which allows us to maximally cluster phonemes without losing the discriminating information. We thereby retain the ability to express speech across languages in a linguistically sound manner.

Sequences of the tokens form the structural features used to identify a language. Most

existing systems predetermine the type (length of sequence) of structural feature to be used regardless of the languages to be classified. At one extreme a limited number of frequently occurring language dependent phonemes or phoneme pairs can be used to discriminate languages. At the other extreme, structural features consist of a large (theoretically infinite) number of relatively rare and long sequences which make up a language.

We believe that the type of structural feature to be used depends on the languages to be identified. Our approach is to customize the feature while keeping the complexity at a minimum. In this thesis we develop a statistically based feature representation and selection. As a result we show that discriminant information in a system of low complexity (small number of features) is contained in sequences of short length but not restricted to single- or pairwise phoneme occurrences.

In this chapter, Section 1.1, will illustrate how a language identification system is typically constructed. In Section 1.2, we will introduce a new approach to automatic language identification.

1.1 From Speech Recognition to Language Identification

Before approaching the subject of language identification, we look at the process of language-dependent speech recognition. Speech is generally recognized in the four stages depicted in Figure 1.1. The acoustic-phonetic function forms the initial stage of speech recognition. It is assumed that a given phrase uttered by different speakers and at different speeds can be maped into the same discrete sequence of tokens. The function can be more or less precise to increase or decrease the speaker dependent variations in the mapping. An imprecise function is therefore more accurate but may not capture the information needed to understand the speech. Thus there exists a tradeoff between accuracy and precision.

During the alignment of speech, the chosen acoustic-phonetic function is used to represent the speech with a string of tokens. The set of tokens to be mapped, usually phonemes, cover the sounds within a language in order to understand the meaning of what has been said. A token set is designed with respect to the necessary detail of representation required by the application. A word spotter for the word "I", for example, would require

only two tokens: "I", and not-"I". On the other hand a full set of phonemes is necessary for understanding the speech.

Detailed structural features can be extracted in the third stage to the degree to which the discrete representation reflects the speech properties needed by the application. These features will be used to understand speech or identify the language of the speech. This application dictates the speech representation and can be used to increase or decrease the complexity of each stage in Figure 1.1. The process of choosing a set of tokens to represent speech, aligning it, and extracting structural features in order to identify a languages will be the focus of our discussion throughout the remainder of this section.

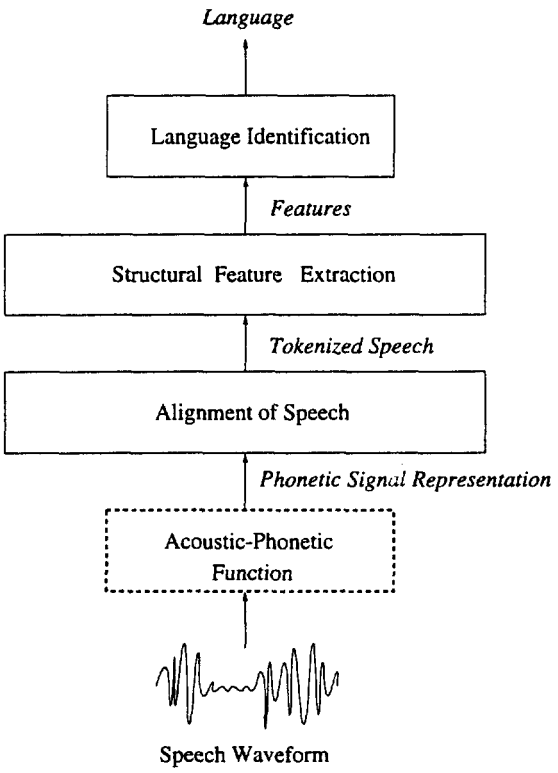

Figure 1.1: Basic processing components of a general speech recognition system.

1.1.1 Acoustic-Phonetic Function

As illustrated in Figure 1.1, the process of identifying a language based on the speech waveform consists of four main steps. The first process of the language identification system is to transform continuous speech into a sequence of discrete events. An utterance of human speech is depicted as a sampled wave form in Figure 1.2. The speech signal reflects the vocal-tract configuration of the speaker. It varies slowly over short periods of time during steady-state vocal-tract configurations ($<$ 100 ms) and changes over longer periods of time as required during production of different speech sounds ($>$ 100 ms). Speech waveforms can thus be segmented into slowly varying sections. The resulting set of segmented intervals of actual sounds are called phones and correspond directly to the different sounds in the language of the speaker.

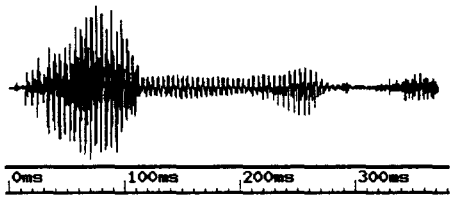

Figure 1.2: Sample wave file of the spoken word "Language"

Phones, in turn, can be mapped into phonemes, the smallest units of speech. Phonemes carry the minimal information needed to distinguish between the words in a given language. Thus, a single phoneme may encapsulate different pronunciations or allophones which are allowed within a language without changing the meaning of a word in which it occurs. An example of this would be the sentence "Boston Harvard Square" uttered by a bostonian only hinting at the /r/. The same sentence produced in the southern dialect of American English would put heavy emphasis on a retroflexed pronounciation of the /r/. Both utterances denote the same sentence. The set of all phonemes occurring in one language is then by definition language dependent. Speech recognition algorithms, using an acoustic-phonetic approach, assume that acoustic patterns can be modeled by a function which is able to map the speech wave to a corresponding phoneme in a given

language.

1.1.2 Alignment of Speech

In Figure 1.3 we can see that the second process towards identifying the language of a given speech wave consists of deriving a set of time aligned discrete events to represent the speech. The description of one possible implementation of the alignment process will be given in more detail in Chapter 5.2. Aligning the continuous speech signal to obtain a sequence of discrete events is the first step towards capturing higher level structural features. It is therefore important to find a set of speech units that captures the amount of appropriate detail for a given application. Figure 1.3 depicts different methods of aligning speech in multiple languages. It displays the spectrum of representing speech with broad categories, such as vowels or consonants, at one extreme, or phonemes at the other extreme. The phoneme-clusters which are introduced in this thesis form a compromise between the two extremes – language-specific phonemes and language-independent broad categories as depicted in Figure 1.3. By clustering phonemes across all languages in the system, we believe that we have gained a linguistically sound way of expressing multilingual speech while maintaining discriminating information.

At one extreme, we distinguish three different ways of using phonemes to represent speech. The state-of-the-art systems (Section 2.1.1) capture speech sounds of all languages in an ALI system by constructing an array of language-dependent phoneme recognizers and aligning the incoming utterance of an unknown language in terms of each of the language-dependent phonemes in a "parallel approach". Phonemes from each language are therefore used in a cross-language manner. Envisioning a system with ten languages, one might consider decoding the speech in terms of only a subset of these front ends. We are now mapping phonemes across languages with the hope of sufficiently covering the space of all occurring speech sounds. In the extreme, phonemes from a single language are mapped across all the languages in the system. Such a system may not be linguistically sound if it discrimininates between languages whose phonemes are not represented in the final system. As an example, let us assume an English utterance is aligned with Japanese phonemes. While English distinguishes between the two phonemes /l/ and /r/,

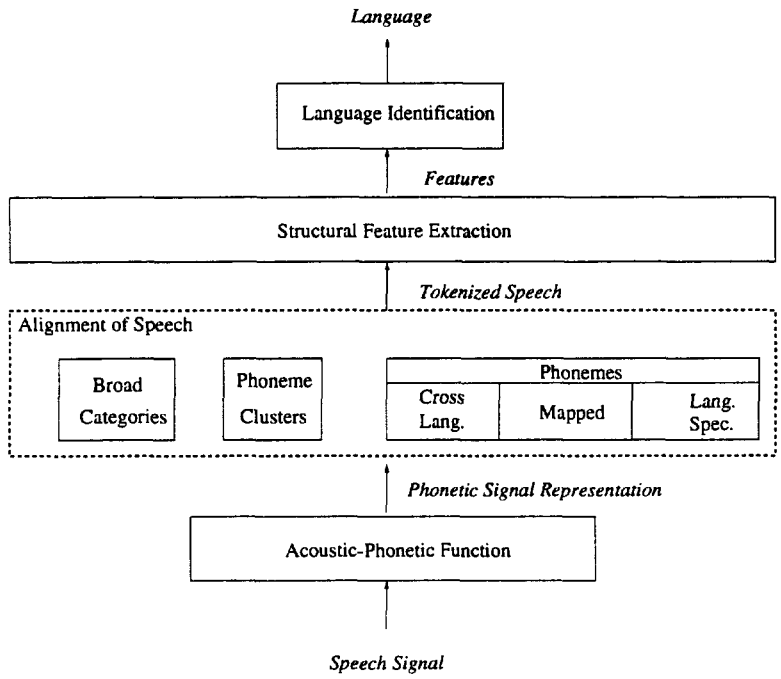

Figure 1.3: Multi-lingual Speech Representation

the Japanese phoneme set will treat both sounds as allophones. This example shows that a language dependent alignment may not be sufficiently complex to capture the sound inventory for several languages.

Alternate methods express multi-lingual speech by creating a supra-lingual speech unit. Creating broad-categories, for example, one can express nasals such as /ng/, /n/, or /m/ in both German and English within a single category. Some languages may differ based on broad categories and their sequences [76]. For example, German, which has sequences of consonants is distinguished from Japanese due to its inherent structure where each consonant is followed by a vowel (consonant-vowel structure) [18]. While broad categories are a linguistically sound representation of multilingual speech (broad categories are valid across languages) they generally discard information which may be useful in distinguish languages. A review of these approaches is presented in Section 2.1.

In the previous section we defined the term phoneme as a language dependent speech

unit. Thus, in order to create a linguistically meaningful speech unit across languages, we introduce phoneme clusters derived from the superset of all phonemes. While phonemes are very precise at expressing speech, the accuracy with which they can be correctly classified is much lower compared to broad categories. For example it is much easier to recognize the category vowel than to identify precisely whether /a/, /e/, or /o/ was said. On the other hand, even though vowels can be recognized with higher accuracy, more precision may be necessary in order to discriminate languages. By clustering vowels, we achieve a compromise between high precision and accuracy by modeling only the discriminating information while creating a speech unit which is able to express multilingual speech.

1.1.3 Structural Feature Extraction

After the continuous speech signal is decoded into a sequence of discrete events, the next subsystem of a speech recognition system generally captures higher level structural features as depicted in Figure 1.4. Language dependent structural properties of speech include phonotactics (which refers to the allowed sequence of phonemes or broad categories within a language). Similarly, syllables (the minimal unit of organization for a sequence of speech sounds, acting as a unit of rhythm. Usually containg a vowel as the nucleus), sub-words and words carry such structural information. Their occurrence can either be language specific or alternatively vary statistically across languages. The level of detail at which the discrete events are represented in the speech signal influences our ability to extract meaningful structural features. This in turn directly affects the performance of the entire system. Section 2.2 will review approaches which incorporate structural features into language identification systems.

1.1.4 Language Identification

Language identification is often used as the front end to a language-specific speech-recognition system. Such a front end is not responsible for understanding the speech. It is therefore not necessary to decode the segmented speech into a string of words and subsequently their meaning as is common in speech-recognition systems. However, we believe that understanding the utterance may ultimately be the key to robustness in language

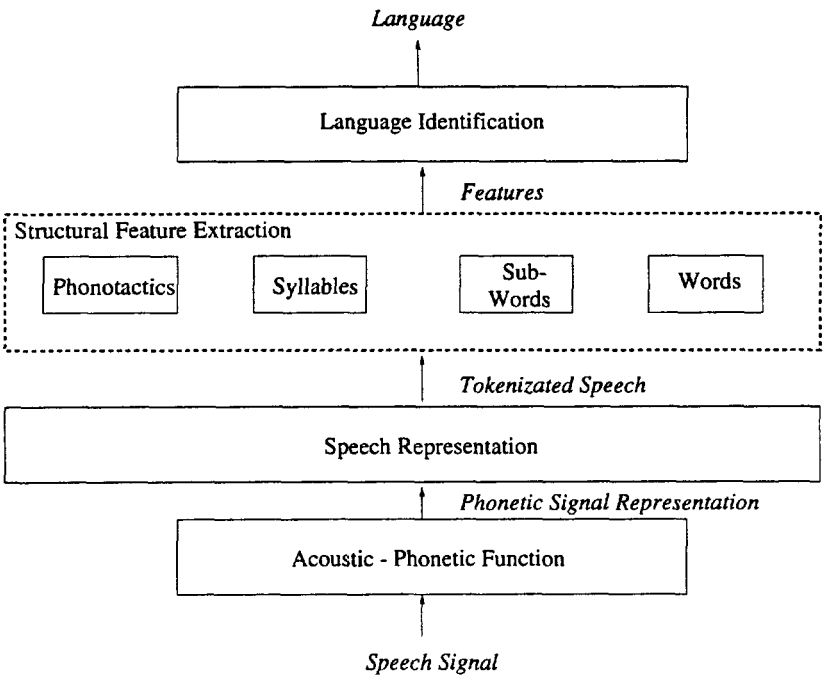

Figure 1.4: The quality of structural features used by the application depends on the speech representation.

identification [82]. For this reason it is important to reduce the complexity of language identification systems in each of its parts to facilitate growth in the number information sources used by the system.

While common approaches to language identification depend only on phoneme and word occurrences, understanding an utterance entails extracting all available features including higher-level language structures. Language-dependent grammars for example denote the systematic structure of a language and capture both the occurrence of language-dependent words as well as their language-specific order within an utterance. Such language models may ultimately disambiguate recognized languages/words/phonemes and thereby improve performance.

While machines still have limited success in incorporating even the most basic information sources efficiently, the hope for the future is to approximate the human process of including as many information sources as possible. If implemented in all its detail,

a speech-recognition system using all these information sources would require immense complexity. Using the parallel approach to language identification, by decoding the utterance with several language-specific phoneme recognizers, and incorporating all possible information sources in parallel, would at the least multiply the complexity by the number of languages in the system. In arguing that this complexity is unnecessary, this research proposes reduced modeling of certain components for a language identification system in order to facilitate the future addition of higher-level information sources.

1.2 A New Approach

The application of the speech recognition system dictates the necessary detail to be modeled in the preceding stages (structural feature extraction and speech representation). Not modeling more detail than necessary is the premise used to simplify the complexity at each stage in our language identification system while retaining a linguistically sound speech representation and features. Structural features are used to discriminate languages. Languages differ in various ways: while some languages may be identified based on a single phoneme, others may be discriminated based only on short words. By not placing any restrictions on the type of features, our system is flexible enough to extract linguistically sound features.

The choice of speech representation is important in order to capture the discriminating information contained in the speech and determines the effectiveness of the structural features for the application. While realizing the importance of precise modeling, we also realize that not all information is equally important. By selectively modeling information which is needed by the application we can benefit from the tradeoff between precision vs. accuracy. In this section we propose a new method that combines a linguistically sound approach to representing speech with statistical methods of extracting language-specific structural features.

After outlining our linguistic reasoning in Section 1.2.1, we motivate the chosen speech representation in Section 1.2.2 and show how we propose to capture structural differences

between languages in order to identify the language of an incoming utterance in Section 1.2.3. Section 1.2.4 outlines the contributions of this thesis.

1.2.1 Linguistic Motivation

The key to a successful implementation of a language identification system which captures language dependent features is twofold. First, a sound representation of speech units must be able to capture the language specific information in a representation customized to the targeted languages. Second, a contrastive feature analysis must capture the differences between languages.

Languages differ from each other in different ways. While some languages may have a similar phoneme inventory (for instance Spanish and Japanese), they may differ at the syllable level. For example, Japanese has a strict consonant-vowel structure while Spanish does allow for consonant clusters at a higher frequency. Root languages such as Chinese and Vietnamese are invariable at the word level, while inflectional languages such as the European languages signal grammatical relationships with word final affixes that are short and frequent. We therefore expect that inflectional languages can in part be discriminated based on shorter sequences while the root languages can be discriminated based on longer sequences. Not only is the length of the sequence important for discrimination but also the level of detail in which the speech units are represented. For example, both German and English are Germanic languages and are therefore similar with respect to allowing consonant clusters, frequency of vowels and grammatical affixes. In such a case detailed phoneme representation of the speech is necessary in order to capture the differences in pronunciation of language specific fricatives, such as the unvoiced dental fricative /th/ in the English word 'Bath' or the uvular fricative /ch/ in the German word 'Bach'. However, some languages, may discriminate based on a much higher level of speech representation. For example, both Chinese and Japanese have a highly constrained syllable structure which can be discriminated at the broad category level.

In Figure 1.5 depicts the four stages of language identification of the input speech wave. In the second stage, the wave form is aligned with a chosen set of speech units. Phoneme clusters are chosen over phonemes and broad categories. This representation allow us to

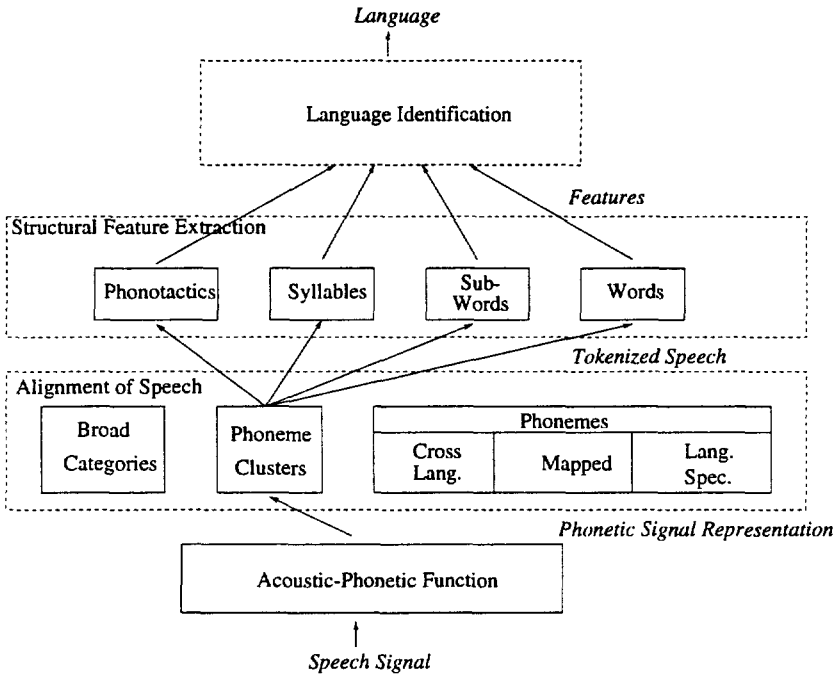

Figure 1.5: Modules of the LID System chosen for this thesis

capture the language discriminating information while reducing the complexity of phoneme modeling and increasing the information from using strictly broad categories. The arrows in Figure 1.5 indicate the modules that are chosen in order to allow both a customized representation and a contrastive feature analysis. One should note that at the level of structural feature extraction this design allows for the necessary flexibility to adapt to the inherent structure of any type of language, while expressing the structure in a flexible phoneme-like representation.

1.2.2 Multi-lingual Speech Representation

Some of the best language identification systems today achieve their performance based on phonemes as the unit of speech. We believe that this amount of detailed modeling may not be necessary [11, 9]. The issue is to show that it is feasible to replace the parallel approach, which models all the phonemes in all the languages, with a more conservative

speech representation. There are a number of reasons why one would prefer to use a less detailed distinction between different speech units, including the following:

- With a growing number of languages, the utterance is decoded in parallel by a larger number of recognizers, thereby increasing the computational cost and decreasing the classification accuracy.
- Decreasing the complexity by choosing a subset of phoneme recognizers to represent all languages in the system may arbitrarily not model important phonemic distinctions.
- With an increasing number of languages, the number of phonemes to recognize proliferates, thereby making it difficult to train a single phoneme recognizer discriminatively across languages.
- The absence of a common phoneme set does not allow cross-lingual statistical analysis of linguistically meaningful phoneme sequences.

The issue is to find the appropriate set of speech units for discriminative training. It should capture sufficient detail to reflect language specific sounds as well as cover the full range of speech sounds from all languages in the system. For this purpose we distinguish between two types of phonemes:

Mono-phonemes	Phonemes occurring in one language
Poly-phonemes	Phonemes similar across languages

As was shown in [8], most of the language dependent information is concentrated in the mono-phonemes and not in the poly-phonemes [1]. Therefore, there is a high degree of redundancy in recognizing speech at the phoneme level with respect to the set of poly-phonemes. In Chapter 5 we uncover this redundancy, by first clustering phonemes across all languages in the system, and then remove it by building a single front end to discriminate between these phoneme clusters.

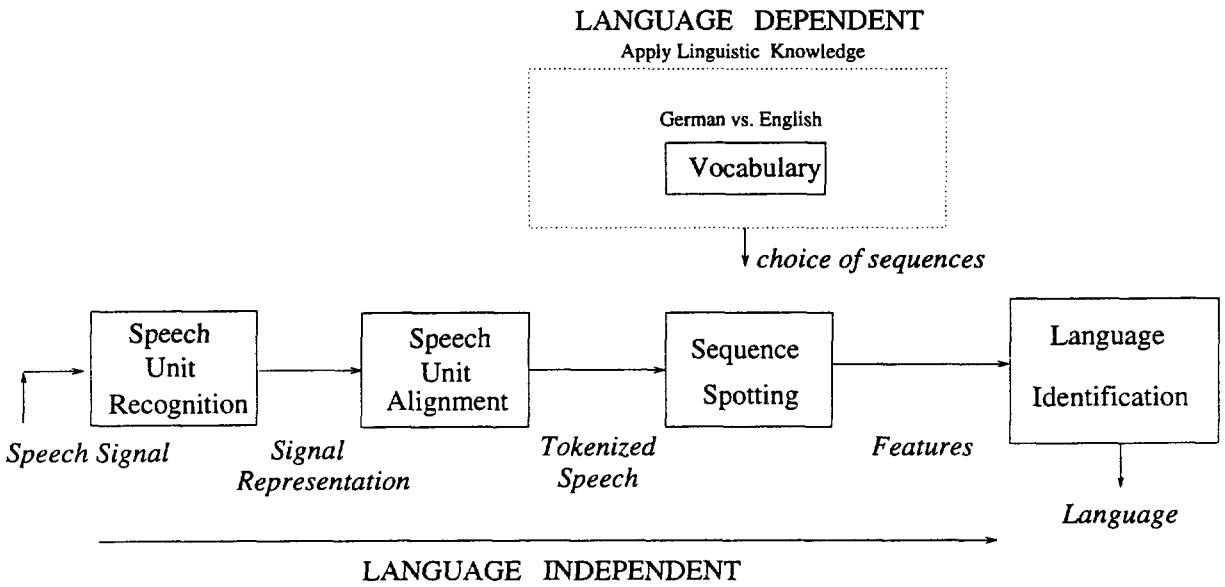

Figure 1.6: Modules of the LID System chosen for this thesis: The system consists of a phoneme recognizer, followed by an automatic alignment of the speech with the recognized phonemes. Finally, features are derived based on the sequences discriminating German and English.

1.2.3 Language Identification

In order to identify the language of an incoming utterance it is passed through a single multi-lingual speech recognizer which aligns speech from all languages with a common set of phoneme clusters. A common set of speech units across languages allows us to apply a single set of structural information sources. Using statistical modeling, we automatically derive discriminating sequences of any length. Each language is then represented by a "Vocabulary" – a set of sequences capturing both its inherent and discriminating structure. Language identification is performed by matching the vocabulary statistics against the words spotted in the incoming speech. The resulting system as shown in Figure 1.6 is implemented to discriminate between English and German as discussed in Chapter 5 and Appendix E.

By ranking features through statistical analysis according to their discriminating power,

[1]Terms coined by Paul Dalsgaard, University of Aalborg, Denmark

it is possible to construct a minimally sufficient set of features for discriminating languages. Through quantitative analysis, by studying the frequency and distribution of linguistic units, regularities governing the structure of language as well the language discriminant features have been elucidated in this manner. This in turn allowed us to reduce the number of features chosen at this level of analysis thereby reducing the complexity in a systematic manner. Chapter 3 will address the development of this theory.

In order to handle within-language variability such as dialects, speaker differences, and bad phoneme recognition, Chapter 4 will present the theory behind a method for extending the "vocabulary" matching algorithm in a structured manner to capture these variablities without losing discriminability between languages. Appendix E will address the practical implementation of this theory.

1.2.4 Contribution

The key contributions of this thesis are threefold: we have combined linguistic design with theoretical derivations and subsequent system implementation.

1. **Original linguistically sound approach to language identification:**
   - representation of multilingual speech with speech units valid across langugages, and
   - flexible feature extraction because languages differ in different ways.
2. **Theoretical derivations:**
   - Introduction of a systematic method for deriving a multilingual set of speech units and for selecting structural features by estimating the discrimination capability of a given system design. For this purpose, a mathematical model is developed to approximate the theoretical discriminability of two classes represented by streams of tokens.
   - Development of a mathematical model to understand the impact of inaccurate alignment on class discriminability. This thesis introduces an argument explaining why using inexact sequence matching to improve language identification

systems has presented a difficult problem to researchers. The implementation and application to multilingual telephone speech verifies the theoretical results.

3. **Implementation:**

- Implementation of the developed theory and verification on automatically generated data.
- Implementation of a language identification module tested on multilingual telephone speech.

Chapter 2

Previous Work

A recent review [77] of language identification contains a detailed literature review of historical and present approaches to language identification. In addition, Muthusamy [76] presents thorough discussions of the subject in his thesis. We will therefore limit our literature review to detailing the modern approaches, and evaluating them with respect to the approach taken in this thesis.

Related work with respect to speech representation will be addressed in Section 2.1. The speech representation used in language identification systems determines the detail with which the structural features are extracted. Research has concentrated on determining a "good" representation with respect to performance (correct discrimination of languages), representing previously unseen languages, dealing with unlabeled data, and extending the number of languages in the system without loss in performance. Early trends in this research area have chosen a language-independent speech representation (broad categories, such as vowel or consonant) which requires little labeled data and is valid across languages. The general tendency in recent years has been towards fine-phonemic modeling of speech resulting in much better performance. Such representation is highly language dependent and requires hand-labeled training data. Research now focuses on using selectively detailed speech modeling in order to reduce the complexity of larger systems and generalizing to unlabeled data.

The related work in structural feature extraction as defined by Figure 2.1 will be addressed in Section 2.2. The set of structural features used by a system can range from simply using phonemes, whose occurrence frequency in the speech indicates the language in which it was spoken, to a large inventory of relatively rare language-dependent words,

whose simple occurrences uniquely identify a language. There is a tradeoff in complexity here. While the simple phonemes are garuanteed to occur in a given amount of speech, that same speech may contain a word from the large-vocabulary system if the corresponding set of words is large enough. The latter system therefore requires a large set of features. Research in recent years focuses on this sort of tradeoff while keeping in mind extensibility to an increasing number of languages (20 and more) without loss in performance.

Section 2.3 will discuss related research areas. While our system focuses on speech representation derived from phoneme labels we also present methods that depend on spectral features and pitch. In addition, text based systems use some of the same features used in identifying speech and are therefore of interest. We can also learn from perceptual studies and compare our features to those used by humans to identify languages.

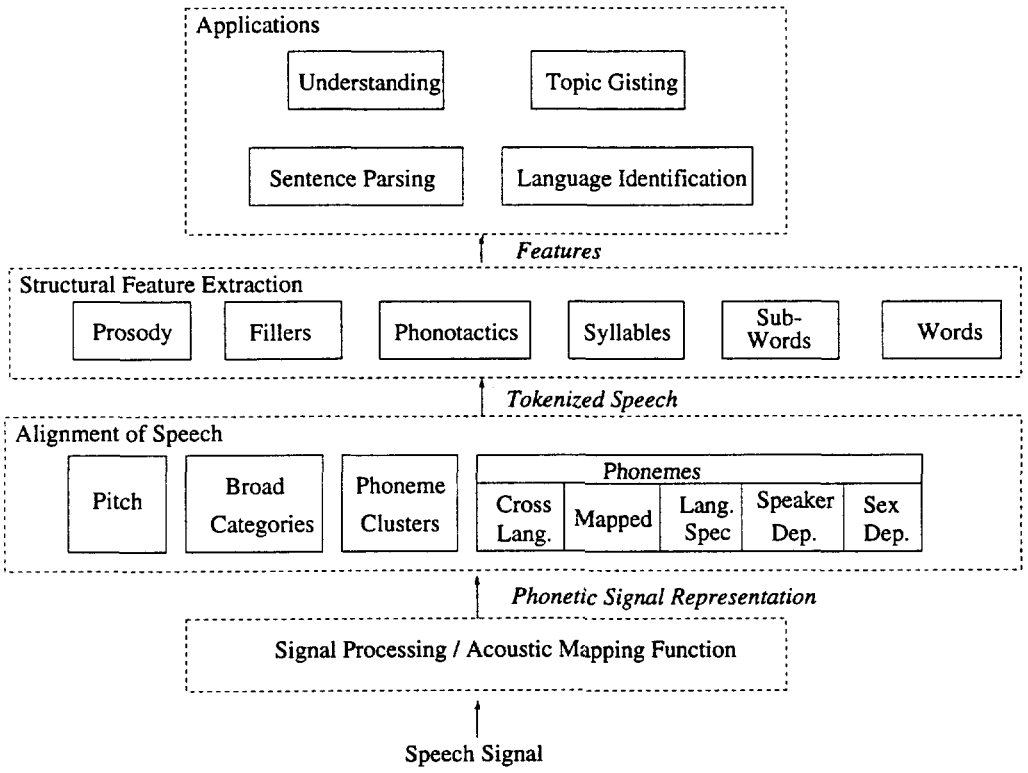

Figure 2.1: Information Sources Used in Multi-lingual Speech Recognition Systems

2.1 Speech Representation

In recent years a number of studies have been performed to find the best speech unit to represent multilingual speech. Systems based on language independent broad categories such as vowels and consonants are easily extensible and generalize to new languages by their lack of detailed information. For this very reason, it was found that detailed phoneme representation captures more language-dependent information and increases language discriminability. However, as the number of languages increases, detailed modeling becomes difficult. The research presented in this section addressed the tradeoff between complexity and performance. We review the general trend from broad category to detailed phoneme representation and indicate another trend from fine phoneme modeling to clustered phonemes as a possible compromise between detailed modeling and broad category speech representation.

2.1.1 From Broad Category to Fine Phonemic Modeling

Increasing the amount of linguistic knowledge contained in the representation of the speech to the recognizers is one of the improvements which allowed the most significant advances in language identification systems. In Muthusamy [78], going from acoustic features to broad categories to phonemes shows large improvements in the system with each step. Work by Zissman et al. and Lamel et al. [111, 113, 57] confirms this issue by moving from ergodic Hidden Markov Models (HMMs) trained on acoustic input to broad-category trained language models to phonemically trained language models.

Virtually all the best language-identification systems today use fine phoneme recognition and language modeling (the exception is the system by K.P.Li based on the spectral representation of speech [64], discussed in Section 2.3). These systems [104, 47, 113, 57] usually have a considerable degree of complexity. Complexity and language identification accuracy can be increased by adding models for additional variabilities such as gender and phoneme duration [56, 57, 112]. I briefly outline some of the important systems that fall into this category.

Lincoln Lab System (1994) [111, 113, 112] HMM-based phoneme recognizers for each of the languages in the system are trained using the phonetically labeled training set from the OGI-TS database. The training speech for each of the N languages is passed through each of the L front end phoneme recognizers, where $N \geq L$. From this stream of phonemes the language model for each of the N languages is trained using bi-gram language models to take phonotactics into consideration. The final likelihood score for each language for each utterance is calculated as the average of the individual log likelihoods emanating from the corresponding language models associated with each channel, see Fig. 2.2.

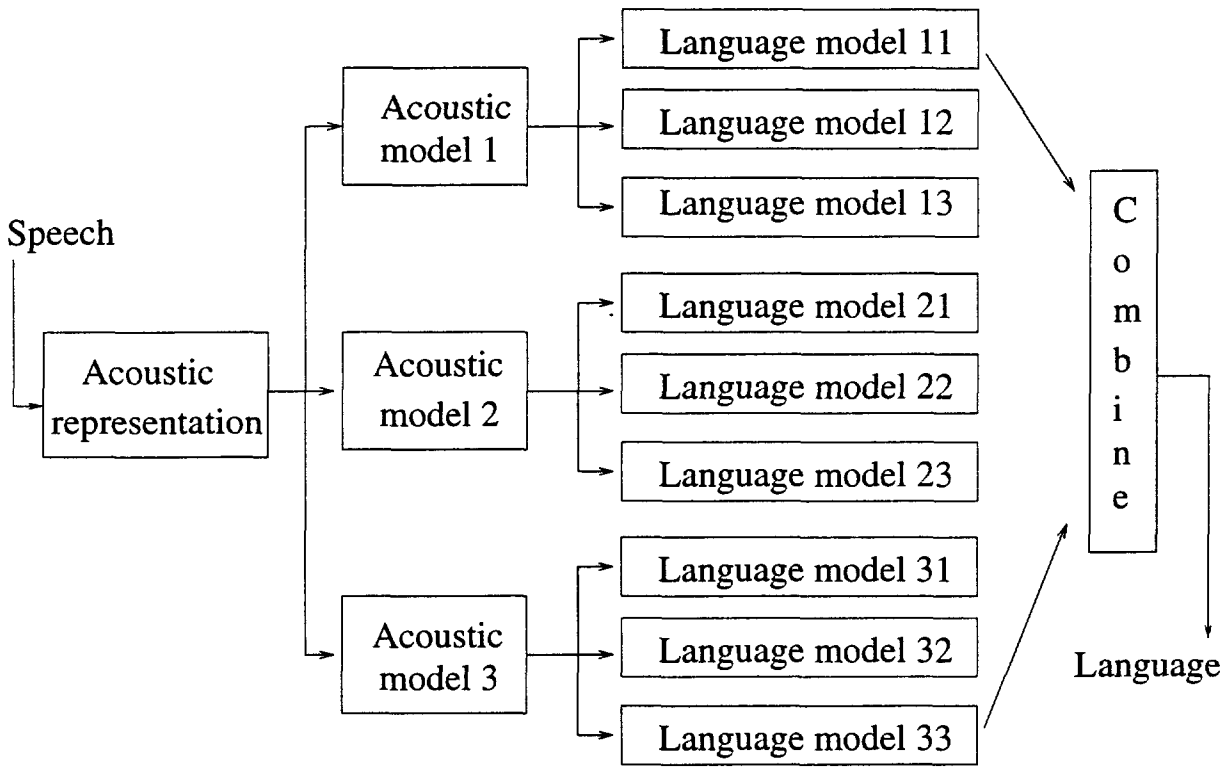

Figure 2.2: Language Identification Using Phoneme Recognition and Phonotactic Language Modeling

To improve this system, it is extended to model duration and gender dependent variabilities.

OGI System (1995) [108, 103, 104, 107, 106] It is believed that increasing the number of parameters increases the variability of the system to the extent that it does not generalize well to previously unseen data. This particular improved system does not increase the number of parameters like the previously mentioned system, but instead seeks to optimize existing parameters.

Starting with a system similar to Zissman's which runs phone recognizers in parallel and trains language dependent models based on each of the phone model outputs, this system contains several important optimizations. Firstly, the language models train not only on the traditional forward bigram probability but also on a backward right-context bigram probability. Secondly, all parameters in the language models are optimized with respect to the squared classification error E:

$$E = (S_T - S_M)^2$$

where S_T is the output score of the target language and S_M is the highest output score not necessarily corresponding to the target language.

Other Systems in this Category Lamel et al. [57] identify a set of non-linguistic speech features in order to model them separately in a recognition system. These include language, gender, speaker, dialect, speech disfluencies, etc. Ergodic HMMs are trained for each of the features and the models are run in parallel during the classification phase. Acoustic likelihoods are computed and the identified language corresponds to the highest scoring model. Lamel mentions that these ergodic HMMs can in theory be trained directly on acoustic data, without the necessity of labeling. In order to automatically label the data, bootstrapping is performed based on a small amount of hand-labeled data, by which the recognizer automatically aligns data using a common set of labels derived from a single language in which the data is labeled. In this manner, Lamel also uses a mapped set of phonemes similar to Zissman's. Two problems with this approach may be the lack of labeled data in some practical situations, and the reduced training data for each of the models.

In order to reduce the complexity, several groups have tried to map phonemes across

languages by representing larger sets of languages in terms of phonemes taken from only a subset of the languages [113, 104]. Extensibility of such systems to a larger set of languages is not clear, and the somewhat arbitrary reduction to phonemes in a specific subset of languages to reduce complexity may no longer be effective as the systems expand. (However, systems today show no degradation in performance when using mapped phonemes for the present tasks).

These systems incorporate very large degrees of statistical linguistic knowledge embedded in the HMMs of phoneme-models and language-models. However, they have not provided much meaningful information about language differences, since those differences are encoded in a large number of parameters. As described in Section 1.1.1, phonemes are not valid across languages. Therefore, expressing one language in terms of the phonemes from another is not linguistically sound. Training language-dependent phoneme recognizers also does not allow discriminant training between phonemes across languages. Most importantly, it is not apparent how these systems will extend to a larger number of languages.

2.1.2 From Fine Phonemic Models to Clustered Phoneme Modeling

The general tendency to move from broad category to fine phonemic modeling, has been supplemented by research which determined that modeling of all phonemes may not be necessary. In Dalsgaard [24], Berkling [8] and Zissman [112] language dependent phonemes, called mono-phonemes or key-phones, have been shown to contain most of the language discriminating information. Dalsgaard uses this knowledge to build a language identification system.

Dalsgaard distinguishes between language dependent phonemes (mono-phones) and language independent phonemes (poly-phones) [24]. By using the similarity of acoustic phonetic features or data-driven clustering (employing a similarity measure based on a global confusion matrix [2, 1]), he either clusters phonemes across languages or identifies them as mono-phonemes. This establishes a super set of labels valid across all languages to label a multilingual database used for training phonemic recognizers [6]. He thus is able to increase the training data for phoneme recognizers by merging poly-phones across

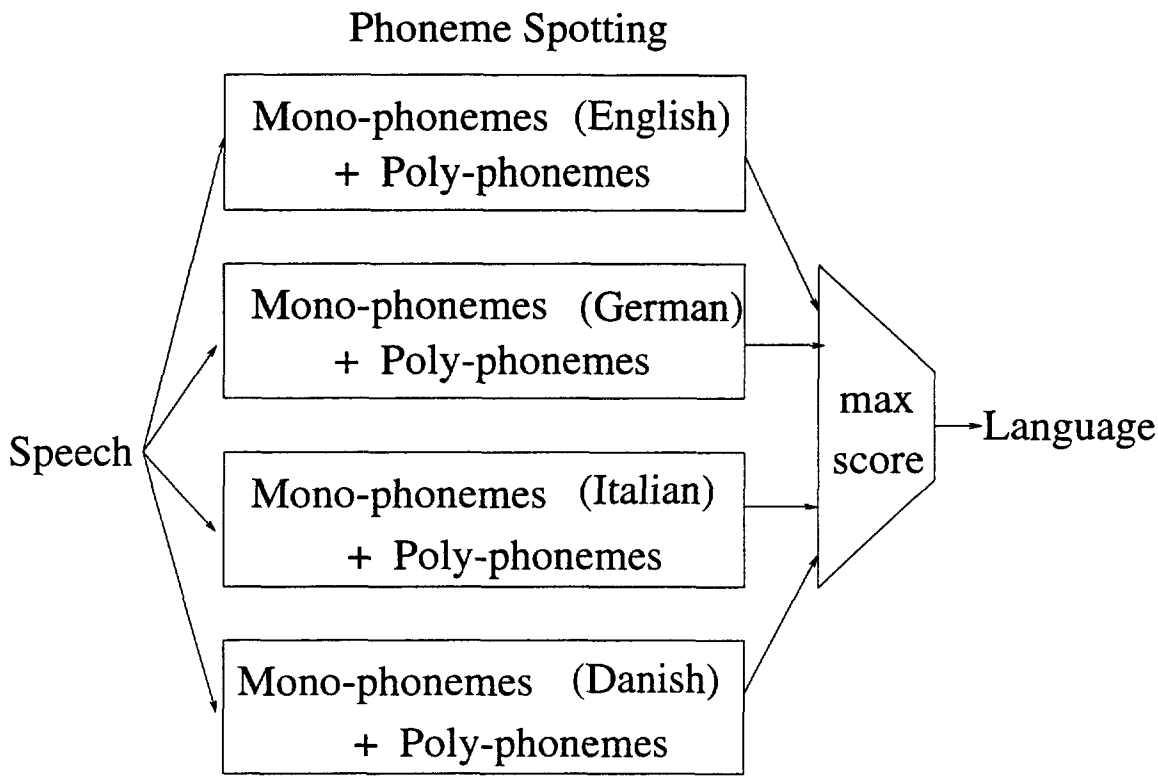

Figure 2.3: Language identification by spotting for mono- and ploly-phonemes. Each subsystem returns a likelihood score. The maximum score identifies the language of the speech.

languages and improving his phoneme recognition in each of the languages. The main result is the ability to substitute poly-phonemes for their language dependent counterparts without losing recognition accuracy on the average.

Dalsgaard employs the mono-phones for language identification by modeling each language in terms of the corresponding mono-phones and the union of all poly-phones across all languages [25]. A grammarless Viterbi search returns an automatically derived alignment of an incoming utterance by using each of the language dependent models. A post-phoneme weighting algorithm emphasizes mono-phoneme occurrences and returns a probability score associated with the alignment. The language model returning the highest score is the language identified. Figure 2.3 shows the system identifying the four

European languages (British English, Danish, German and Italian).

This system is based on a *minimally clustered* common set of labels across languages. In other words, if a phoneme /b/ is realized in all four languages in a similar enough manner, /b/ becomes a poly-phoneme merging training data from all the languages. In contrast, *maximal clustering* would promote clustering of phonemes not only across but also within a language. The similarity measure between phonemes to be clustered is relaxed to allow for example /p/ and /b/ to merge. This is a step leading back towards broad category speech representation. The reasoning behind such a merge lies in recognizing that the merge does not decrease language discriminating information while increasing training data for this phoneme class. Dalsgaard's use of poly-phonemes unioned across all languages can potentially increase the complexity when extending the system to say twenty or more of the world's languages. It is also not clear how many languages a poly-phoneme can represent. In fact, as the number of languages increases the number of poly-phonemes can still potentially grow unmanageably large. This may possibly be avoided by clustering maximally to increase training data while maintaining pairwise high language discriminability.

2.2 Structural Feature Extraction

Structural features of a language can be modeled at several levels. Using a simplified view, the first level might analyze speech in terms of phonemes, syllables, or word occurences. A second level would arrange the features in the first level by using a language dependent grammar with the goal of understanding the speech. Language identification typically differs from speech recognition by ignoring the second level in this simplified scheme. Modern systems are characterized by the type of feature that is extracted from the tokenized speech: features often correspond to bigram and trigram probabilities (capturing the phonology of a language) or a large set of words. In all systems the type of feature is predetermined independently of the language in the system.

2.2.1 Language Identification with Word/Sequence Spotting

While the preceding examples have all revolved around phoneme recognition, other approaches which extend the systems to include word level information also exist. While one can argue that the use of phone recognition is more efficient and task independent than the use of words, experience with monolingual recognition systems has shown that adding information at the word level can increase the robustness of a system with respect to language identification. The use of sub-word modeling maintains task independence. It is clear that incorporating word level recognition directly into the complex systems described above would be a formidable task. This section outlines some approaches taken where word or sub-word modeling is used.

Triphone Modeling In Kadambe [47] the sub-word models consist of triphones. The assumption here is that triphones are optimal sub-word models regardless of the languages in the system. In going from context independent to context dependent phoneme modeling, more linguistic knowledge can be applied and therefore an additional source of discriminative features is captured. Since this procedure decreases the amount of training data available per parameter, emphasis is put on increasing the available training data via usage of phoneme sequences derived from textual sources. Triphone probability is traditionally [45] estimated as follows:

$$Pr(s_3|s_1, s_2) = \lambda_3 f(s_3|s_1, s_2) + \lambda_2 f(s_3|s_2) + \lambda_1 f(s_3),$$

where s_i denotes phoneme symbol i, $f()$ denotes the frequency of occurrence, and λ_3, λ_2, and λ_1 are set to 1, 0, and 0 respectively. The probabilities $Pr(s_3|s_1, s_2)$ are summed over the entire utterance using parallel language-dependent phoneme alignment. The language subsystem with the highest log likelihood is chosen as the language of the input speech signal.

This system is extended to include lexical modeling. In this case the incoming utterance is processed in parallel by language-dependent phone modeling followed by the corresponding language model and lexical model. Lexical models each contain 2000 words which are unique to that language. Good results are obtained due to the increased amount

of linguistic knowledge now in the system. While aligning languages A and B with the phoneme set from language A, the assumption is that language-specific characteristics are retained. The obtained results confirm this assumption. However the high complexity required of such a system may possibly be avoided, by selectively choosing sub-words according to their discriminating power between languages.

Embedded Word Models of Frequent Words and Phrases The system by Ramesh and Roe [87] is based on the general design used in all of the above systems. Their word-spotting algorithm depends on the existence of specific words in the utterance to be classified since words are specified at the phoneme level. However, they hope to improve in some cases but never lose performance by adding word models. As pointed out earlier, one of the problems with using word spotting is that the process becomes topic dependent.

2.2.2 Topic Identification with Word/Sequence Spotting

Gish et al. approached language identification by applying algorithms developed originally for topic identification [31, 93, 70, 65] The solution is decomposed into three subtasks: (1) Keyword selection, (2) topic modeling, and (3) event detection. Keywords are selected based on a score which calculates how much each word contributes to the discrimination between any two given topics. The two distance measures used are the Kullback-Leibler distance and the mutual information measure. The topic is modeled by training a linear neural-network classifier based on occurrence counts of selected sequences taking advantage of occurrence dependencies among the selected keywords. An incoming utterance is classified by spotting for the given sequences and passing their occurrence frequency to the trained network and classifying according to maximal likelihood. In an incoming utterance the number of keyword events is estimated by summing the probability of a putative hit over time. The probability at time t of having seen the keyword is calculated using the forward-backward scoring algorithm used in the Baum-Welch algorithm. Gish et al. [65] apply this algorithm to language identification by generating pseudo-word clusters. Each cluster is represented by the cluster centroid in the form of a single sequence of English

phonemes. Pseudo-word spotting is performed by finding inaccurate matches. The identified language corresponds to the maximal log likelihood of matching the incoming string to two language models in the pairwise system.

A similar approach to topic identification developed by Moore and Novell [84], is based on spotting for unusual and topic dependent words. These words tend to be long, so a word spotting algorithm needs to take into account errors due to both the errorful phoneme transcription and variations in ponunciations. This algorithm can be applied directly to language identification, even though, rather than looking for rare, long words one now selects frequent and mostly shorter pseudo-words. Other work closely related to this approach is recent work done at Ensigma relating to topic identification and language identification [16, 85, 101].

The above approaches are all related to one another and were developed during the same time as the sequence of research which forms the core of this thesis [78, 108, 8, 11, 9, 10]. We also base language recognition on keyword selection, detection and classification. The language classification in this thesis is non-linear and the weighted keyword selection is based on the Bhattacharyya distance. Furthermore, sequences in [65] can only belong to one cluster, whereas for our system they can be associated with several clusters. Both algorithms were developed independently and the approaches differ significantly because, in addition, this thesis deals with the cross-lingual phoneme complexity.

2.3 Related Methods

When tokenizing the speech into a set of phoneme-based labels other information in the signal is often discarded. Pitch and spectral features fall into this category. Such features have been reported by subjects participating in perceptual studies in language identification. It is therefore important to look at related research on human perception of languages. Other features identified by humans, relate to frequently occuring sequences of speech sounds that are typical for a given language as well as rare language dependent words. Such a feature set has been the subject of Section 2.2 and is also addressed in text-based speech recognition.

2.3.1 Language Identification without Phonemes

Pitch can be an important feature when captured correctly. This has repeatedly been reported by human listeners during perceptual experiments [82]. While this is an important feature, we consider pitch to be an orthogonal issue; pitch (like other information sources of information such as grammar and speech understanding) can be added to existing systems.

Pitch. Pitch has been shown by Itahashi to be useful when basing the identification exclusively on this feature [42, 43]. Itahashi uses prosody as the sole feature in his language identification system. He argues that fundamental frequency is more robust than segmental parameters in noisy environments. The most interesting result to note from this work is that he is able to distinguish between three Asian languages (Chinese, Korean, and Japanese) and three European languages (German, French, and English) by using parameters derived from ratios of occurrence frequencies of the pitch slopes.

Hazen [37] on the other hand was less successful when using pitch, prosody, and duration as a standalone system, or when adding pitch to his existing language identification systems. Hazen integrates the prosodic model with acoustic modeling, a phonological language model, and a-priori language probability. Optimal scaling factors are required to integrate these features appropriately. However, the effect of the prosodic model in terms of performance is not remarkable.

Spectral Features. Since speaker-dependent differences can be greater than language-dependent differences, K.P. Li [64] models speech in speaker- and language-dependent dimensions. It can be shown that the variation in the spectral features between any pair of speakers is much larger than the language differences obtained from a single bilingual speaker. The key is to choose features that reduce the differences between two speakers in order to emphasize the language differences. Li uses a marking of syllabic nuclei which is both language- and speaker- independent. These syllabic nuclei are used to calculate features which represent a given utterance. Each language is represented by a set of representative speakers within that language. A minimum score refers to the minimal

distance between a test utterance and a reference utterance. The language corresponding to the matched speaker is returned. Alternatively, the language whose top N speakers match the incoming utterance best on the average can be returned. This may take care of dialect differences, channel changes, and outliers. Combining both methods results in best performance of the system. The main point to note in this approach is the language independent feature extraction in terms of the syllabic nuclei. In contrast, the phonemic approach and the probability of bigrams are inherently language dependent at the phonemic level, and thus would create problems when performing the segmentation process in a language independent manner.

2.3.2 Other Related Work

Language identification is also related to research beyond the confines of conventional speech processing. We limit our discussion to perceptual studies and text-based systems.

Perceptual Experiments Taking part in the perceptual experiment [82] had one of the greatest effects on the way we designed our system. Our own experience, and those of others voiced during interviews after the test, were that language identification was rarely based on long words but mostly on short syllables and phoneme occurrences. In addition, perceived tone of voice (harsh, deep, high) was important, as well as pitch and intonation across and within phones. Our clustering approach is based on the fact that there is only a small set of phonemes that are language dependent, and most other phones do not contribute significantly to the listener's language identification process unless they occur in specific context.

Text-based Systems Ziegler [109] built a text-based language-identification system. He employs occurrence frequencies of signals ("subwords") much as we propose. Ideal properties of signals are high frequency and significant inter-language variability. Ziegler therefore uses a weighting for scoring which is based on signal detection according to linguistic significance. This is analogous to the clustering done in this thesis where emphasis is placed on such speech units which occur in one language only, or Dalsgaard's emphasis

on mono-phonemes as described in Section 2.1.2. Ziegler's system is fast and accurate. He incorporates linguistic knowledge and most importantly is able to process a very large set of languages (150) with high accuracy in an efficient manner.

Most current systems which employ lexical access as an additional module to the spoken language identification system rely on a critical mass of lexical words which tend to be long, numerous and rare across both length and topic of speech. Grammatical Morphemes on the other hand make up the shorter signals which are highly frequent within a language. The drawback of working with these types of signals for speech recognition systems is the high detection error rate and the risk of deletion of such subwords from speech. In addition, such short sounds may not be sufficiently distinguishable across languages due to the error rate of the recognizer. Such problems may however in part be overcome by training discriminatively on speech units across all languages and selecting sequences in the same manner. Most of the top discriminating sequences obtained by taking both recognition accuracy and their frequency of occurrence into account are still at the short morpheme level.

Chapter 3

Feature Modeling and Discrimination of Languages

One of the main contributions of this thesis is the mathematical theory underlying the design of a system for language identification based on discriminating sequences of speech units. This theory will be the subject of the next two chapters. While Chapter 4 will explore the effects of misrecognitions on language discriminability, this chapter will derive the theoretical model of the features used for discrimination and an estimate of the language identification error.

In order to enable a cross-lingual statistical analysis of sequences, used to discriminate languages, a set of speech units which is meaningful across languages is developed in Section 3.1. Speech units are derived by clustering phonemes across languages in order to minimize the necessary modeled detail without loosing the essential language specific information. Thus we take advantage of the tradeoff between modeling precision and accuracy of recognition. However, it is not feasible to implement a language identification system for each level of possible clustering. A theoretical estimate is therefore developed in order to determine whether a merge of phonemes across or within languages decreases the ability to discriminate between the languages in the system.

Section 3.2 and Section 3.3 develop a method for estimating the discrimination error based on a given set of labels. Training data aligned at the maximal level of precision is automatically relabeled at the proposed clustered level. Based on this data, language dependent features can be modeled using a normal distribution. This model takes into account the mean occurrence frequency of a given feature, the variation across speakers

and the variation due to the length of the available speech. Using this model discriminating features can be extracted by estimating the Bayes' error due to two language dependent distributions. For each of the features the corresponding discrimination error is estimated and the top N features combined in order to indicate the performance of language discrimination based on this chosen set of speech representation.

In an iterative process described in Section 3.4 the language identification error is estimated for a given set of phoneme clusters in order to derive the optimal set, balancing accuracy vs. precision without losing language discriminating information. Section 3.5 tests the theoretical error estimate on automatically generated data in order to validate the use of an estimate for reducing the precision of the speech representation.

3.1 Speech Unit derivation

Clustering of phonemes across languages is based on the premise that not all phonemes are of equal importance to the language identification task. In fact, decreasing the number of phonemes to be recognized may improve the phoneme recognition accuracy which in turn may improve alignment (as described in Chapter 1.1) and, therefore, language identification. We start with the union of all phonemes from the languages to be recognized and cluster these to obtain a reduced set of phonemes. Construction of a clustering tree concerns both the order of merging as well as the termination of merging, i.e. the pruning of the tree. In our approach, merging relates to the *closeness* of two phoneme classes, whereas the pruning depends on the decrease in theoretically estimated language identification. Each merge of phoneme classes should satisfy the following two requirements

1. The performance of the alignment should improve.

2. Language classification should not deteriorate due to a merge.

In order to guarantee an increase in performance of phoneme recognition, we chose the information theoretic mutual information distance measure. If we view the phoneme recognizer as a channel between the acoustics and the Viterbi search as described in Section 1.1.1, then we want this channel to carry a maximum amount of information

about the incoming signal. Information is highest when the input is most difficult to guess, and the output correctly reflects the input.

Let $p(x|y)$ be the conditional probability of recognizing y as x after alignment. With $p(y)$ denoting the prior probability of y, $p(x) = \sum_y p(y)p(x|y)$ is the estimated occurrence frequency of x after alignment. The expected mutual information is then given by:

$$MI = \sum_{x,y} p(y)p(x|y)ln\left[\frac{p(x|y)}{p(x)}\right] \tag{3.1}$$

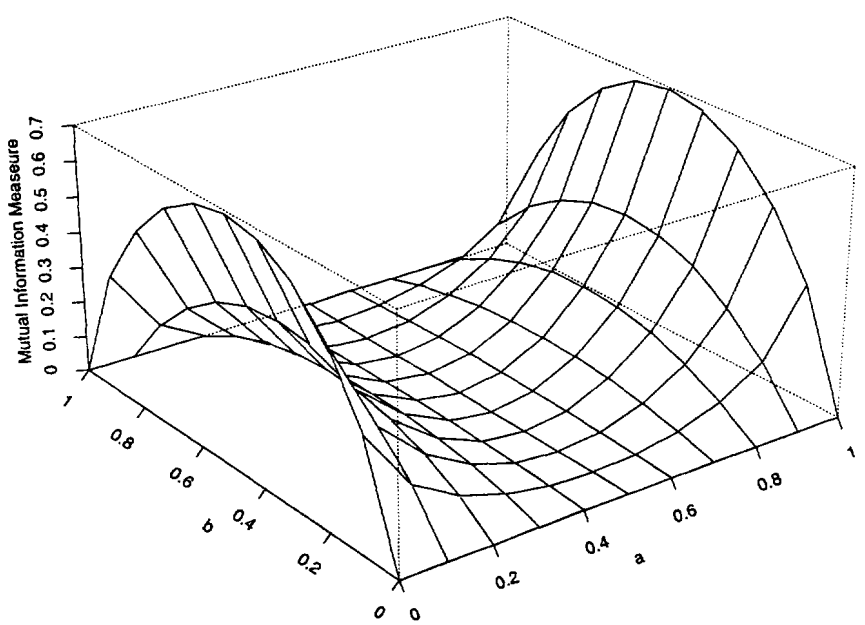

Figure 3.1: Effect of confusion matrix on expected mutual information. **b** and **1-b** denote the prior probability of the two occurring classes. **a** is defined in Equation 3.2

To improve our understanding of this formula, we look at a two class problem. Here, the confusion matrix, denoting the probability of choosing x given y, is assumed to have the form:

$$p(x|y) = \begin{pmatrix} a & 1-a \\ 1-a & a \end{pmatrix} \tag{3.2}$$

and the prior probabilities $p(0) = b$ and $p(1) = 1 - b$. Figure 3.1 shows the value of the mutual information measure as a function of a and b as given above. As can be seen, the expected mutual information is largest when the priors are evenly distributed and confusion between classes is low. Increasing the mutual information measure while merging phoneme classes therefore decreases confusion between the classes while keeping the priors distributed as evenly as possible.

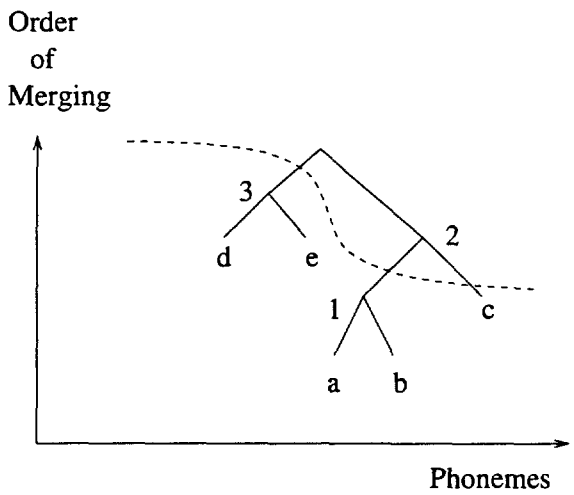

Figure 3.2: Example clustering tree showing the sequence of merges from phonemes to phoneme clusters

At each merge two phonemes are chosen which result in the maximal increase in the mutual information measure. Before merging phonemes across languages, an estimate of the language identification error is obtained. Subsequent merging of phonemes are allowed with the constraint that original discriminability does not decrease. As an example, Figure 3.2 shows a sequence of merges as indicated by the numbers 1..3. Suppose merge

2 would have increased the estimated language classification error and is therefore disallowed, while merge number 3 does not affect language identification error. This means that merge 3 can potentially increase phoneme-cluster recognition performance and will be merged even though it is at a higher level than merge 2. The final pruned tree for this hypothetical example is indicated by the dashed line.

3.2 Feature Modeling

In order to know when a merger is disallowed, it is essential to have a good estimate of the language identification error. The discriminability of two languages is based on a model of the features used to identify a language. Based on such a model the similarity or dissimilarity of two languages can be quantified for each feature. These can then be sorted and selected accordingly. The features used in this thesis correspond to occurrence frequencies of sequences of labels. Such a sequence L can consist of one or more labels. Consider a string of labels which was hand labeled by a human, where L corresponds to the sequence to be examined and x is any other event: **L x x x x L L x x L L L**. In this example, the occurrence frequency of L, $f(L)$, in the labeled string is $6/12 = .5$. A language is represented by N native speakers. Each of the speakers' wave file has been handlabeled with a string of labels. The goal of this section is to model the frequency of occurrence of L in the set of labeled strings from one language. We assume a normal distribution of sequence occurrences within a language, and model the two parameters u and s (corresponding to the mean and standard deviation of the frequency distribution), of any sequence L in language $\mathcal{L}$. We write

$$f[L] \in N(u_{\mathcal{L}}[L], s_{\mathcal{L}}[L]) \tag{3.3}$$

to signify that, across all utterances in language $\mathcal{L}$, the occurrence frequency of string L is normally distributed with parameters $u_{\mathcal{L}}[L]$ and $s_{\mathcal{L}}[L]$.

In order to model the distribution of $f[L]$ in a short sample we have to account for

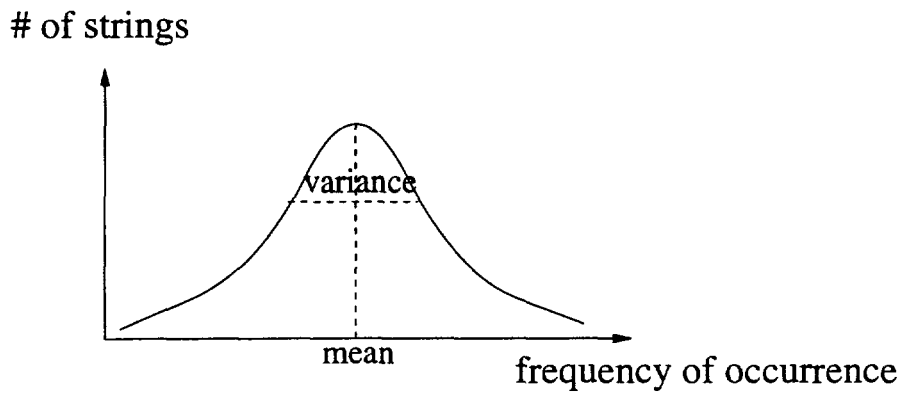

Figure 3.3: Normal Distribution.

all the factors that render it variable. We distinguish between inter- and intra-speaker variances. The former ($s1$), is the variation due to different phoneme usage by different speakers and the variation due to speaker dependent phoneme recognition. The latter ($s2$), is a function of the length of a given speech sample. A phoneme sequence L is useful for discriminating between languages $\mathcal{L}_i$ to the degree that these variabilities are smaller within a language than across languages. We now develop a model for the two random processes that determine the variance of occurrence frequency for a given sequence in a speech sample such that the final variance of word L in language $\mathcal{L}$ is due to both s_1 and s_2.

$$\begin{aligned} f \ &\epsilon \ N(u_{\mathcal{L}}[L], s_{\mathcal{L}}[L]) \\ &= N(u_{\mathcal{L}}[L], \sqrt{(s1_{\mathcal{L}}[L]^2 + s2_{\mathcal{L}}[L]^2)}) \end{aligned} \tag{3.4}$$

Variance Due to the Speaker: s1 The variance $s1$ is due to different speakers and different topics for various speech samples. This variance is also affected by the speaker dependent phoneme recognition. We assume normal distribution across all speakers here. This variance will be constant for all lengths of speech samples. Let N be the number of strings or speakers. Let $f_{\mathcal{L}}[L]_q$ denote the frequency of word L for speaker q in language $\mathcal{L}$. The mean $u_{\mathcal{L}}[L] = \frac{1}{N}\sum_{q=1}^{N} f_{\mathcal{L}}[L]_q$ is calculated in the conventional manner. The variance

$s1_{\mathcal{L}}[L]$ is then given as follows,

$$\begin{aligned} s1_{\mathcal{L}}[L]^2 &= Var(f_{\mathcal{L}}[L]) \\ &= \frac{1}{N-1}\sum_{q=1}^{N}(f_{\mathcal{L}}[L]_q - u_{\mathcal{L}}[L])^2 \end{aligned} \tag{3.5}$$

Variance Due to Length of Speech: s2 The variance $s2$ is due to the length of the given speech sample. We assume that the normal distribution arises as the limit of a binomial distribution. This variance will go to zero as time goes to infinity. To have a clearer understanding of why we chose a binomial distribution to model this variance, assume we are looking for a label or sequence L in a string of labels. At any time t, after having seen t segments, there exist t slots which can potentially be labeled L. Let x denote the number of segments that are labeled L , where $(x < t)$. Then $\frac{x}{t}$ is the normalized occurrence frequency of L at time t. Suppose further that u is the mean occurrence frequency of L as $t \to \infty$. Then the variance of the occurrence frequency due to the short time sample t is:

$$\begin{aligned} var(L) &= E[(\frac{x}{t} - u)^2] \\ &= \sum_{x=0}^{t}(\frac{x}{t} - u)^2 p(x)n(x), \end{aligned} \tag{3.6}$$

where $p(x)$ is the probability that a specific sequence containing x sequences labeled L will be obtained, and $n(x)$ is the total number of such sequences:

$$p(x) = u^x(1-u)^{t-x} \tag{3.7}$$

and

$$n(x) = \binom{x}{t} \tag{3.8}$$

The actual variance is therefore a weighted sum.

$$\sum_{x=0}^{t} \binom{x}{t} u^x (1-u)^{t-x} (\frac{x}{t} - u)^2 \tag{3.9}$$

Thus, expressing this formula in terms of the parameters already used in previous equations, let N be the number of labeled strings corresponding to different utterances. $u_{\mathcal{L}}[L]$ denotes the mean occurrence frequency of word L in language $\mathcal{L}$. With t corresponding to the number of segments seen, the variance $s2_{\mathcal{L}}[L]$ is given by,

$$\begin{aligned} s2_{\mathcal{L}}[L]^2 &= \textstyle\sum_{x=0}^{t} \begin{pmatrix} t \\ x \end{pmatrix} u_{\mathcal{L}}[L]^x (1 - u_{\mathcal{L}}[L])^{t-x} (\tfrac{x}{t} - u_{\mathcal{L}}[L])^2 \\ &\simeq u_{\mathcal{L}}[L]^2 (1 - u_{\mathcal{L}}[L])^2 e^{-t u_{\mathcal{L}}[L]} \tfrac{(t u_{\mathcal{L}}[L])^t}{t!} \\ &\Rightarrow \lim_{t \to \infty} s2_{\mathcal{L}}[L]^2 = 0 \end{aligned} \tag{3.10}$$

Fig. 3.2 shows the rate at which $s2_{\mathcal{L}}[L]$ converges to zero as time t goes to infinity and mean frequency $0 \leq u_{\mathcal{L}}[L] \leq 1$. One sees that this component of the variance can be quite sizeable if u is much larger than zero.

In Section 3.5 this model is validated on an artificial problem, and in Section 5.5.1 it is compared with the performance on real speech data.

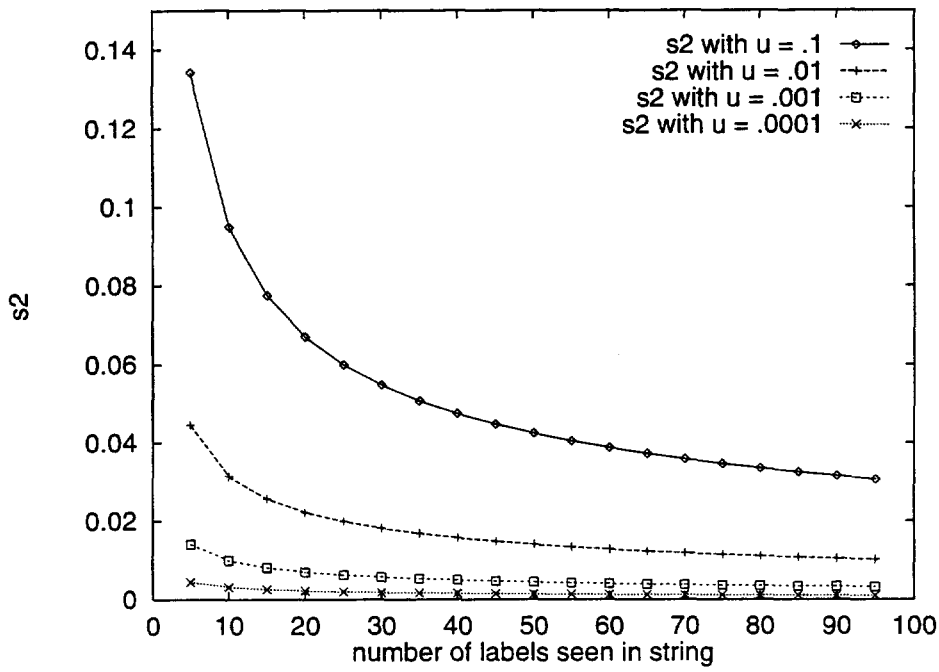

Figure 3.4: s2 as a function of time (in segments) for different values of u.

3.3 Feature Selection

The classification error between two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ due to a single sequence L can be estimated based on the model of the occurrence frequency of L in the respective languages developed in the previous sections. Given the assumption of normal distributions, it is possible to calculate the Bayes error from measured means and variances for the different languages. This is, however, overly precise in light of the assumptions that were made during the modeling, such as the assumptions of normal distribution and other regularities that may not occur in real speech. A simpler measure is adopted which allows us to perform analytic calculations without a degradation in accuracy. This choice is verified in the Chapter 5 where the theoretical estimate is matched to actual data. Given the distribution of sequence occurrences as derived in the previous section, the discrimination error for a pair of languages based on a one-dimensional feature space of one sequence is estimated with the Bhattacharyya distance [30] as given here.

$$error = \frac{1}{2} e^{-\frac{1}{4}[\frac{(u_2[L]-u_1[L])^2}{s_1[L]^2+s_2[L]^2}] + [\frac{1}{2} log \frac{1}{2} \frac{s_1[L]^2+s_2[L]^2}{s_1[L]s_2[L]}]} \tag{3.11}$$

Suppose now, that we want to estimate the joint error due to any two sequences resulting in a two dimensional feature space. Along each dimension the mean and variance corresponding to the chosen sequences with normal distributions. Assuming independence of features, we next derive a vector $\vec{\alpha}$ which will be used in the weighted scalar product with the feature vector to produce a linear mapping onto a single dimensional space as depicted in Figure 3.3.

$$\begin{aligned} \mu &= \alpha_1 u[1] + \alpha_2 u[2] \\ \sigma^2 &= \alpha_1^2 s[1]^2 + \alpha_2^2 s[2]^2 \end{aligned} \tag{3.12}$$

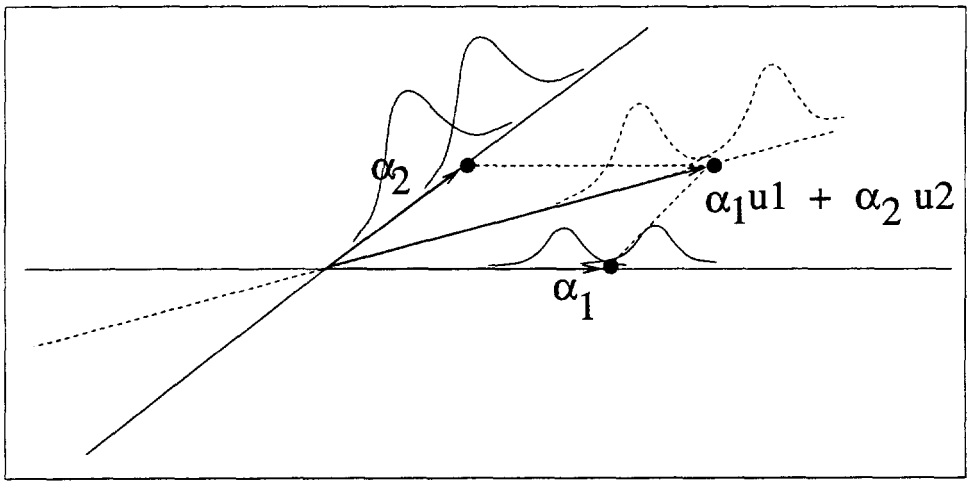

Figure 3.5: Combining two features linearly for optimal discrimination.

The goal is to derive the vector $\vec{\alpha}$ which minimizes the error. In order to find the optimal α, the error will be differentiated with respect to $\vec{\alpha}$. If we were to use the Bayes error, this would result in a non-linear equation, to be solved numerically. Thus, we will approximate $\vec{\alpha}$ using the Bhattacharyya distance. Since differences in mean occurrence frequency are

more pronounced than differences in variances, the second term in Equation 3.11 can be dropped, u replaced by μ, and s replaced by σ, resulting in:

$$error = \frac{1}{2} e^{-\frac{1}{4}[\frac{(\mu_2[L]-\mu_1[L])^2}{\sigma_1[L]^2+\sigma_2[L]^2}]} \tag{3.13}$$

This simplification is verified for real data in Chapter 5. We calculate the optimal weight $\vec{\alpha}$ by taking the derivative of the error with respect to $\vec{\alpha}$ and setting it to zero to solve for $\vec{\alpha}$ at the minimum error. $\mu_{\mathcal{L}}[j]$ and $\sigma_{\mathcal{L}}[j]$ as given in Equation 3.12 are expanded into an equation which can be solved for $\vec{\alpha}$. Without loss of generality we can set $\alpha_1 = 1$. Then the following derivation solves for α_2

$$\frac{\partial}{\partial\alpha_2}error = \frac{\partial}{\partial\alpha_2}\left[\frac{1}{2}e^{-\frac{1}{4}[\frac{(\mu_2-\mu_1)^2}{\sigma_1^2+\sigma_2^2}]}\right] = 0$$

$$\Rightarrow \frac{\partial}{\partial\alpha_2}\left[-\frac{1}{4}[\frac{(\mu_2-\mu_1)^2}{\sigma_1^2+\sigma_2^2}]\right] = 0$$

$$\Rightarrow -\frac{1}{2}\frac{\mu_2-\mu_1}{\sigma_1^2+\sigma_2^2}(\frac{\partial\mu_2}{\partial\alpha_2}+\frac{\partial\mu_1}{\partial\alpha_2}) + \frac{1}{4}\frac{(\mu_2-\mu_1)^2}{(\sigma_1^2+\sigma_2^2)^2}(\frac{\partial\sigma_1^2}{\partial\alpha_2}+\frac{\partial\sigma_2^2}{\partial\alpha_2}) = 0$$

$$\Rightarrow \frac{1}{2}(\frac{\partial\mu_2}{\partial\alpha_2}-\frac{\partial\mu_1}{\partial\alpha_2}) = \frac{1}{4}\frac{\mu_2-\mu_1}{\sigma_1^2+\sigma_2^2}(\frac{\partial\sigma_1^2}{\partial\alpha_2}-\frac{\partial\sigma_2^2}{\partial\alpha_2})$$

$$\text{Since } \sigma_i^2 = s_i[1]^2+\alpha_2^2 s_i[2]^2$$

$$\text{and } \mu_i = u_i[1]+\alpha_2 u_i[2]$$

$$\Rightarrow u_2[2]-u_1[2] = \frac{1}{2}\frac{\mu_2-\mu_1}{\sigma_1^2+\sigma_2^2}(2\alpha_2 s_1[2]^2+2\alpha_2 s_2[2]^2)$$

$$\Rightarrow u_2[2]-u_1[2] = \frac{\mu_2-\mu_1}{\sigma_1^2+\sigma_2^2}\alpha_2(s_1[2]^2+s_2[2]^2)$$

$$\text{Let } \triangle u[i] = u_2[i]-u_1[i]$$

$$\text{Let } S[i]^2 = s_1[i]^2+s_2[i]^2$$

$$\begin{aligned}\text{then } \mu_2-\mu_1 &= (u_2[1]+\alpha_2 u_2[2])-(u_1[1]+\alpha_2 u_1[2])\\ &= (u_2[1]-u_1[1])+\alpha_2(u_2[2]-u_1[2])\\ &= \triangle u[1]+\alpha_2\triangle u[2]\end{aligned}$$

$$\Rightarrow \triangle u[2] = \frac{\triangle u[1]+\alpha_2\triangle u[2]}{S[1]^2+\alpha_2^2 S[2]^2}\alpha_2 S[2]^2$$

$$\Rightarrow \triangle u[2]S[1]^2+\alpha_2^2\triangle u[2]S[2]^2 = \alpha_2\triangle u[1]S[2]^2+\alpha_2^2\triangle u[2]S[2]^2$$

$$\Rightarrow \alpha_2 = \frac{\triangle u[2]S[1]^2}{\triangle u[1]S[2]^2}$$

$$= (\frac{u_2[2]-u_1[2]}{u_2[1]-u_1[1]})(\frac{s_2[1]^2+s_1[1]^2}{s_2[2]^2+s_1[2]^2})$$

Figure 3.6: Derivation of α

The weighting of the new word is proportional to the difference in mean occurrence frequency between the two languages. However, the higher the combined variance of the occurrence in the two languages is, the smaller the weighting. α_2 is therefore intuitively a measure of goodness of the new word in comparison to the first word. Given the optimal $\vec{\alpha}$ the estimated error between the two languages is guaranteed to decrease after adding the second word.

$$\frac{1}{2}e^{-\frac{1}{4}[\frac{(\mu_2-\mu_1)^2}{\sigma_1^2+\sigma_2^2}]} \leq \frac{1}{2}e^{-\frac{1}{4}[\frac{(u_2[1]-u_1[1])^2}{s_2[1]^2+s_1[1]^2}]} \tag{3.14}$$

In order to estimate the error based on a list of N sequences the top N sequences in the aligned strings from both languages, sorted by their estimated error according to Equation 3.13 [1], are chosen. Rather than optimizing all $\alpha[j]$ at the same time, we choose to calculate a suboptimal solution by optimizing each $\alpha[j]$ separately. Let $\mathcal{L}$ denote the language and let $List(N)_{\mathcal{L}}$ be the overall distribution of the list of N sequences representing language $\mathcal{L}$. $\vec{\alpha}$ is chosen to minimize the error between

$$List_{\mathcal{L}_1} \simeq N(\mu_{\mathcal{L}_1}, \sigma_{\mathcal{L}_1}) \;\; and \;\; List_{\mathcal{L}_2} \simeq N(\mu_{\mathcal{L}_2}, \sigma_{\mathcal{L}_2})$$

Thus we have,

$$\begin{aligned} List(N-1) &\simeq N(\mu'_{\mathcal{L}}, \sigma'_{\mathcal{L}}) \\ Word(N) &\simeq N(u_{\mathcal{L}}[N], s_{\mathcal{L}}[N]) \\ List(N)_{\mathcal{L}} &\simeq N(\mu'_{\mathcal{L}} + \alpha u_{\mathcal{L}}[N], \sqrt{\sigma'^2_{\mathcal{L}} + \alpha^2 s_{\mathcal{L}}[N]^2}) \\ List(N)_{\mathcal{L}} &\simeq N(\mu_{\mathcal{L}}, \sigma_{\mathcal{L}}) \end{aligned} \tag{3.15}$$

Global optimality of the vector $\vec{\alpha}$ is not the goal at this stage. Instead we seek a prediction measure which will compare two different sets of labels with each other. Using

[1] From here on we will refer to this simplified equation as the Bhattacharyya distance.

the same suboptimal solution for both will result in a quantitative measure which allows me to choose one label set over the other. Classification optimization will be done only in the end, after the best label set was chosen.

In this section an error estimate for a given sequence was derived. Based on this value, a list of discriminating features consisting of sequences could be derived for which in turn the language identification error can be estimated. It is now possible to determine whether a set of speech units that is used across all languages in the system is sufficiently detailed to express the differences between languages.

3.4 The Complete Algorithm

Using the theory developed in the previous four sections we can now formulate an iterative algorithm to estimate the discrimination error between two languages based on a list of optimally selected words represented in terms of a given set of labels. There are three essential steps in this process:

1. Frequency Modeling of Sequences.
2. Sequence Selection and Error Estimation.
3. Phoneme Merging.

The flowchart shown in Figure 3.7 depicts the iterative process of merging phonemes and estimating language classification error. At the extremes of this algorithm we obtain either a phoneme based system [36] or a broad category based system [76]. This algorithm will return the lowest number of phoneme clusters without losing language discriminability, together with an error estimate as a function of time due to the chosen features.

The **input** to this algorithm will be both the labeled and the automatically aligned training files at the phoneme level, the hierarchical clustering algorithm developed in Section 3.1, and the recognition accuracy of the phoneme recognizer in form of a confusion

matrix [2]. The algorithm will estimate the language identification error based on a list of sequences at each level of phoneme clustering, as discussed in this chapter. Based on this estimate it is determined whether phonemes may be clustered without losing the ability to discriminate between the languages in the system. The **output** will be the maximally clustered phonemes along with an estimate of maximum error as a function of the length of the input speech.

[2] If labeled files are not available, one may consider using the aligned files and a confusion matrix based on a self-similarity measure (for example the distances between the HMM models) to calculate the expected mutual information.

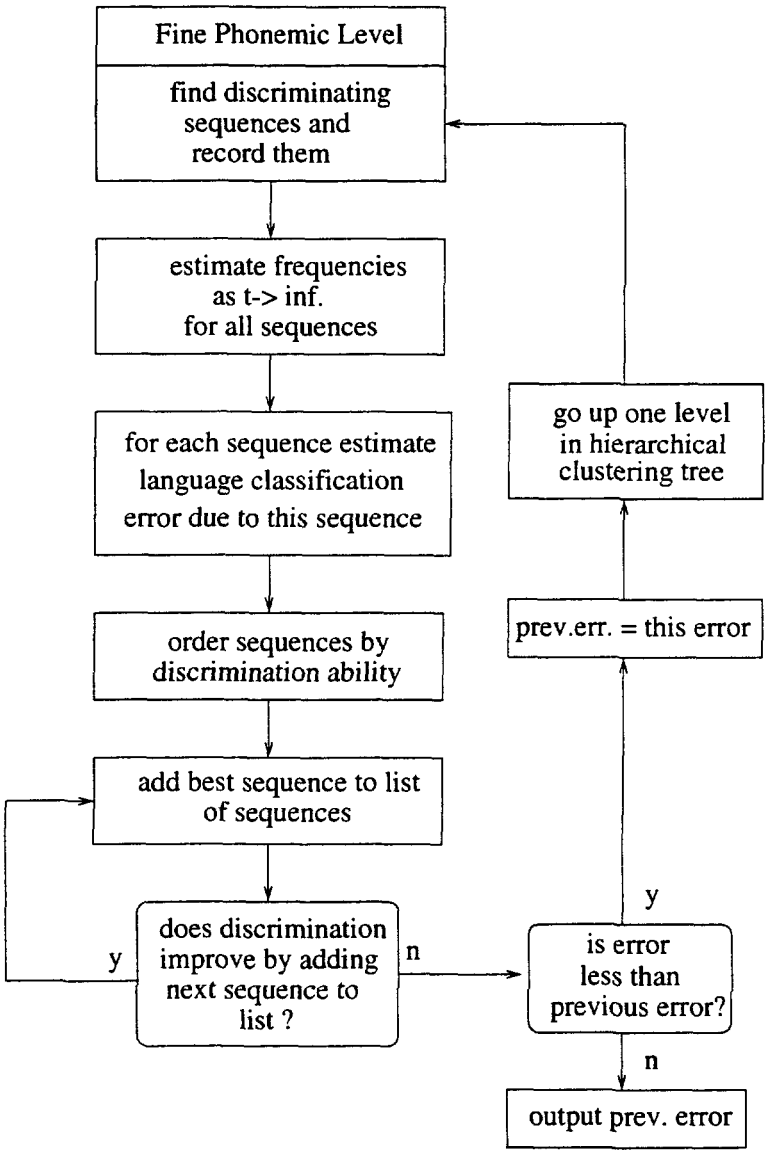

Figure 3.7: Flowchart for estimating the error of an LID system and deriving the appropriate set of labels.

3.5 Language Classification

A string of unknown language is classified by matching features that were extracted from a training set. Features correspond to the top N sequences chosen from the training set, modeled with parameters $\sigma_1, \sigma_2, \mu_1, \mu_2$ as shown in Equation 3.15. Let $\vec{f}$ be the vector of length N where each element of the vector denotes the frequency of occurrence of the corresponding feature (sequence) at time t in the string of the unknown language. Let y denote the scalar product of $\vec{\alpha}$ and $\vec{f}$; then the string of labels is classified to belong to language $\mathcal{L}_i$ according to the maximum likelihood as shown in Equation 3.16.

$$Language = \underset{\forall i \in lang}{argmax} \frac{1}{\sqrt{2\pi}\sigma_{\mathcal{L}_i}} e^{-\frac{1}{2}[\frac{y-\mu_{\mathcal{L}_i}}{\sigma_{\mathcal{L}_i}}]^2} \tag{3.16}$$

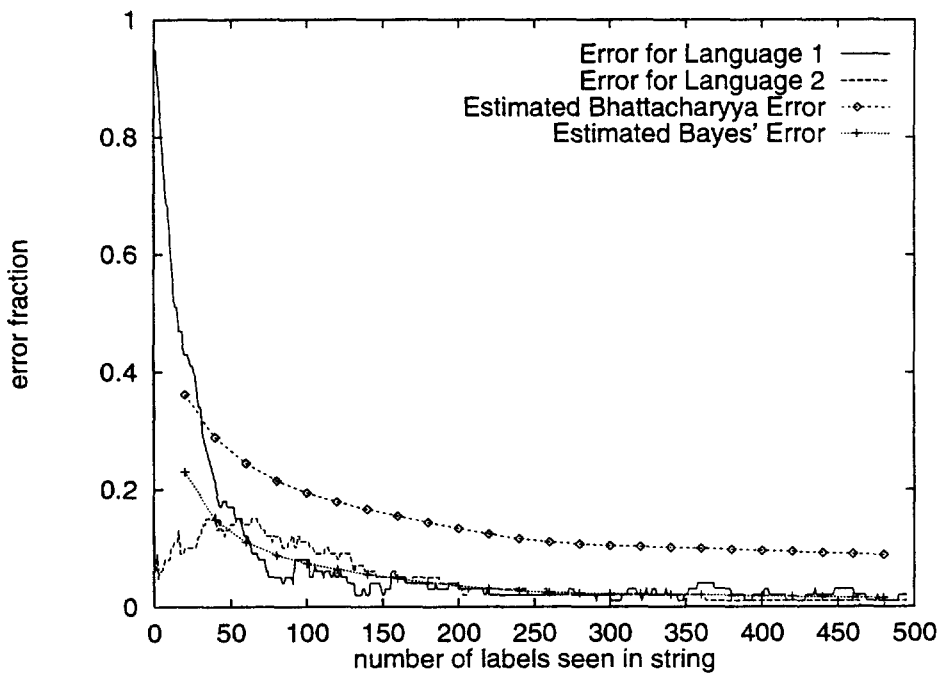

Figure 3.8: Actual error from model language τ and estimated errors from Bhattacharyya distance and Bayes' error. Plot shows fractional error as a function of time (measured in terms of the number of phonemic segments observed).

The classification depends on statistics computed based on all labels seen up to and

including the present label at time t. Reclassification of a string takes place each time a new label is seen and the statistics are updated. The error is a function of the variances and therefore decreases as $s2$ goes to zero. Fig. 3.8 shows how the estimated errors compare to the actual error rate when classifying the model language τ as specified in Table 3.1. (see Chapter 4.1.1). It can be seen that the Bhattacharyya distance provides a fairly loose bound on the actual error. This may actually be more appropriate on real data since some of the assumptions of independence that were made may not be accurate in speech as they were in this model language.

Table 3.1: Example of specifications for feature parameters. Assuming normal distributions, only the mean and standard deviation have to be specified.

$\mathcal{L}_1(i) \epsilon N(u_1(i), s_1(i))$	: $\mathcal{L}_1(a) \epsilon N(0.05, 0.01)$
	: $\mathcal{L}_1(b) \epsilon N(0.02, 0.01)$
	: $\mathcal{L}_1(c) \epsilon N(0.04, 0.01)$
$\mathcal{L}_2(i) \epsilon N(u_2(i), s_2(i))$	: $\mathcal{L}_2(a) \epsilon N(0.01, 0.01)$
	: $\mathcal{L}_2(b) \epsilon N(0.01, 0.01)$
	: $\mathcal{L}_2(c) \epsilon N(0.01, 0.01)$

Chapter 4

Effect of Inaccurate Alignment on Language Discrimination

The model described in Chapter 3 will now be extended to include the effect of inaccurate alignment on the discriminability of languages. The goal is to model the distribution of the aligned data based on knowledge of the labeled data and the faulty alignment process. The inverse of the alignment process and the model of the aligned data can in turn be used to reestimate the accurate labeled data. The hope is to use this information to improve language discriminability by improving the chosen set of features.

We will model the output of three classifiers using labeled, aligned and reestimated labeled data in order to compare their performance. The first classifier is based on true data with no misrecognition (analogous to labeled real-world speech data), the second is based on data with misrecognitions (analogous to automatically aligned speech data). A third classifier is based on a new method which uses inexact sequence spotting in order to reestimate the true data from the aligned data and knowledge of the alignment process. Section 4.1 will introduce the specification for the model language and will define the notation used in the rest of this chapter. The model language τ is constructed in order to simulate the theoretical derivations.

The output distributions of the three classifiers are derived based on the theory developed in Chapter 3. Section 4.2 will extend this model by including the characteristics of the misrecognitions during the alignment process. The impact of inaccurate alignment on language discrimination can then be derived theoretically. In order to compensate for the misrecognitions in the aligned data we want to reestimate the accurate labeled data.

Using the inverse of the alignment process and the modeled misrecognitions of the aligned data, Section 4.3 introduces a method of inexact sequence matching. We show, however that this attempt to reduce the adverse effects of the alignment process is not effective in improving discrimination of languages.

4.1 Modeling True Data

Before studying the effect of automatic alignment on the degradation of features for language identification, this section will first develop the model for specifying feature distributions in the true data. True data is analogous to hand-labeled data in real speech which is assumed to be the correct representation of speech. Given the specifications and the model developed in the previous chapter, we are able to develop a notation used to derive the corresponding parameters, the mean μ and the variance σ^2.

4.1.1 Specifying the Model Language τ

A generator for τ creates N strings of labels whose parameters u and s for normal distribution $N(u, s)$ are specified for two languages. An example of the distributions of any number of labels in the true data of languages $\mathcal{L}_1$ and $\mathcal{L}_2$ (i.e. with no misrecognitions, similar to hand-labeled data in a real language) was given in Table 3.1. Any number of strings can then be generated to represent both languages reflecting the specified distributions. In this example, there are three labels, (a,b, and c) that occur in each of the languages, but with higher frequency in $\mathcal{L}_1$. In general labels a,b, and c will be used to denote relevant sequences in the true data and labels x,y, and z will be used to describe the corresponding sequences in the aligned data. Without loss of generality, these labels are place holders for sequences of any length. Misrecognitions due to the alignment will be described in Section 4.2. Note that these probabilities are not expected to add up to 1.0. We assume there exist other sequences which are not used for language identification. This allows us to assume independence of occurrence frequency for each sequence.

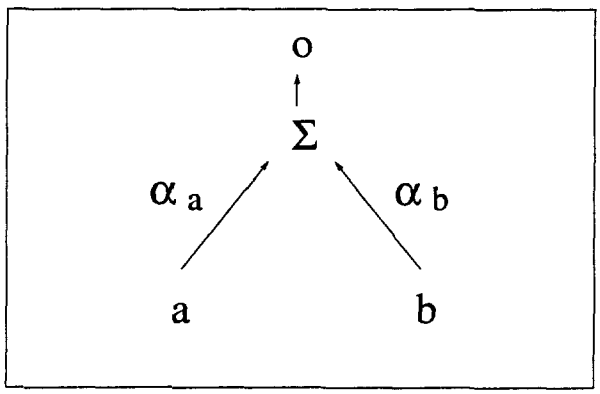

Figure 4.1: Channel with recognition probabilities.

4.1.2 Language Discrimination

Based on the specifications, the distribution of the occurrence frequencies of each feature in the different languages is modeled according to the theory developed in Chapter 3.2. The specified features are then combined according to Chapter 3.3 using a weighting vector $\vec{\alpha}$. Discrimination of two languages depends on the degree of overlap between the two language dependent distributions of the classifier output o shown in Figure 4.1.

In order to estimate the discrimination error due to the two language dependent distributions of o we derive the mean μ_o and the variance σ_o for the distributions of the classifier output. Given the definitions in Table 4.1, the expected value of the output is derived as follows: If symbol a occurred n_a times, b occurred n_b times, the expected number of occurrences of o is

$$\begin{aligned} E[n_o|n_a n_b] &= E[\sum_{i=1}^{n_a} \alpha_a + \sum_{i=1}^{n_b} \alpha_b] \\ &= n_a \alpha_a + n_b \alpha_b \\ \Rightarrow E[n_o] &= \alpha_a E[n_a] + \alpha_b E[n_b] \\ \Rightarrow \mu_o &= E[n_o] \\ &= \alpha_a u_{\mathcal{L}_i}[a] + \alpha_b u_{\mathcal{L}_i}[b] \end{aligned} \tag{4.1}$$

Table 4.1: Definitions

n_x	number of labels x in the aligned data
n_y	number of labels y in the aligned data
n_a	number of labels a in the true data
n_b	number of labels b in the true data
ϕ^i_{ax}	a binary random variable denoting whether the ith occurrence of label a in the true data goes to x.
p_{ax}	$E[\phi^i_{ax}]$
μ_a	mean occurrence frequency of label a
σ^2_a	variance of occurrence frequency of label a

Similarly,

$$\begin{aligned} E[n_o^2|n_a n_b] &= E[(\sum_{i=1}^{n_a} \alpha_a + \sum_{i=1}^{n_b} \alpha_b)^2] \\ &= E[\sum_{i=1}^{n_a}\sum_{i=1}^{n_a} \alpha_a + \sum_{j=1}^{n_b}\sum_{j=1}^{n_b} \alpha_b + 2\sum_{i=1}^{n_a}\sum_{j=1}^{n_b} \alpha_a\alpha_b] \\ \Rightarrow E[n_o^2] &= \alpha_a^2 E[n_a^2] + \alpha_b^2 E[n_b^2] + 2\alpha_a\alpha_b E[n_a n_b] \end{aligned} \tag{4.2}$$

Assuming independence:

$$\begin{aligned} E[n_o^2] &= \alpha_a^2 E[n_a^2] + \alpha_b^2 E[n_b^2] + 2\alpha_a\alpha_b E[n_a][n_b] \\ &= \alpha_a^2(u_{\mathcal{L}_i}[a]^2 + s_{\mathcal{L}_i}[a]^2) + \alpha_b^2(u_{\mathcal{L}_i}[b]^2 + s_{\mathcal{L}_i}[b]^2) + 2\alpha_a\alpha_b u_{\mathcal{L}_i}[a]u_{\mathcal{L}_i}[b] \end{aligned} \tag{4.3}$$

Thus,

$$\begin{aligned} \Rightarrow \sigma_o^2 &= E[n_o^2] - E[n_o]^2 \qquad \text{(by definition)} \\ &= s_{\mathcal{L}_i}[a]^2\alpha_a^2 + s_{\mathcal{L}_i}[b]^2\alpha_b^2 \end{aligned} \tag{4.4}$$

The classifier for this model has already been developed and modeled on generated data in the previous chapter in Section 3.5. The goal of this chapter is to extend this model to include the alignment process.

4.2 Modeling Data with Misrecognitions

In order to classify the language of an utterance, its time signal (waveform) is generally automatically time aligned with a string of labels as described in Chapter 1.1. The resulting string does not agree perfectly with the string of labels created by a human expert for the same utterance which we assume to agree with the intended string. The confusion matrix P is used to specify the relation between true data and the aligned data with misrecognitions as depicted in Figure 4.2. While features in the true data were assumed to be independent, due to the confusion matrix this may no longer be true for the features in the data after alignment. Distributions for both cases are derived and the impact of the alignment process on language discrimination is discussed.

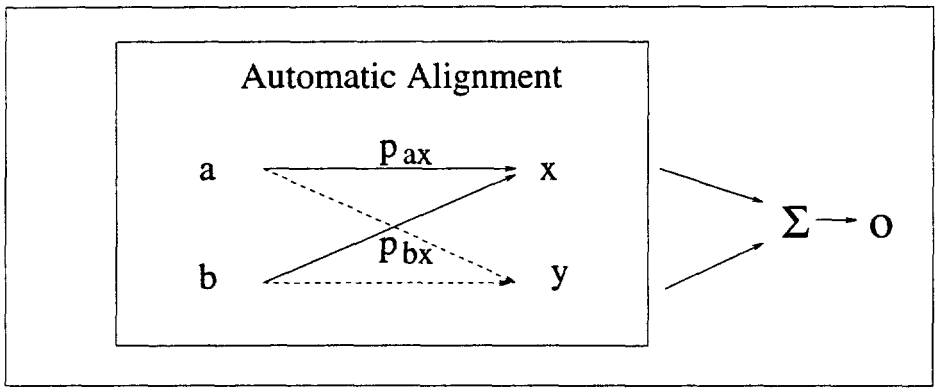

Figure 4.2: Misrecognition probabilities due to alignment.

4.2.1 Modeling Independent Features

Before modeling the case of interdependent features, it is useful to first look at independent features. A confusion matrix P can then be specified in the model describing the recognition performance of the alignment process (the "channel"). An example of P which retains feature independence is given in Table 4.2. The frequency of occurrence $\tilde{u}_{\mathcal{L}}[X]$ of any sequence X in the aligned data for language $\mathcal{L}$ relates to the actual occurrence $u_{\mathcal{L}}[A]$ of the sequence A in the true data as follows:

Table 4.2: Confusion matrix due to alignment with independent recognition of labels.

	aligned x	aligned y	aligned z	other
true a	0.8	0.0	0.0	0.2
true b	0.0	0.1	0.0	0.9
true c	0.0	0.0	0.6	0.4
other	0.0	0.0	0.0	1

$$\tilde{u}_{\mathcal{L}}[X] = p_{AX} u_{\mathcal{L}}[A] \tag{4.5}$$

The variance depends on p_{AX} accordingly,

$$\begin{aligned}
\tilde{s}1_{\mathcal{L}}[X]^2 &= Var(\tilde{u}_{\mathcal{L}}[X]) \\
Var(\tilde{u}_{\mathcal{L}}[X]) &= E[(p_{AX} u_{\mathcal{L}}[A] - p_{AX} f_{\mathcal{L}}[A])^2] \\
&= E[(u_{\mathcal{L}}[A] - f_{\mathcal{L}}[A])^2 p_{AX}^2] \\
&= p_{AX}^2 E[(u_{\mathcal{L}}[A] - f_{\mathcal{L}}[A])^2] \\
&= p_{AX}^2 Var(f_{\mathcal{L}}[A]) \\
&= p_{AX}^2 s1_{\mathcal{L}}[A]^2
\end{aligned} \tag{4.6}$$

The distribution of the occurrence frequency for sequence X in the aligned data is then used as the estimate for distribution of sequence A in the true data. The resulting normal distribution is described by mean $\tilde{u}$ and variance $\tilde{s}$.

$$\begin{aligned}
\tilde{f}_{\mathcal{L}}[X] \ \epsilon \ & N(\ \tilde{u}_{\mathcal{L}}[X]\ ,\ \tilde{s}_{\mathcal{L}}[X]\ ) \\
&= N(\ p_{AX} u_{\mathcal{L}}[A]\ ,\ \sqrt{\tilde{s}1_{\mathcal{L}}[X]^2 + \tilde{s}2_{\mathcal{L}}[X]^2}\ ) \\
&= N(\ p_{AX} u_{\mathcal{L}}[A]\ ,\ \sqrt{p_{AX}^2 s1_{\mathcal{L}}[A]^2 + \tilde{s}2_{\mathcal{L}}[X]^2}\ )
\end{aligned} \tag{4.7}$$

where $\tilde{s}2_{\mathcal{L}}[X]$ is obtained from Equation 3.10 by replacing $u_{\mathcal{L}}[A]$, the actual frequency of A in the true data with $\tilde{u}_{\mathcal{L}}[X]$, the frequency of X in the aligned data.

$$\begin{aligned}\tilde{s}2_{\mathcal{L}}[X]^2 &= \sum_{x=0}^{t} \binom{t}{x} ((\tilde{u}_{\mathcal{L}}[X])^x (1-(\tilde{u}_{\mathcal{L}}[X]))^{t-x} (\frac{x}{t} - (\tilde{u}_{\mathcal{L}}[X]))^2 \\ &= \sum_{x=0}^{t} \binom{t}{x} ((p_{AX} u_{\mathcal{L}}[A])^x (1-(p_{AX} u_{\mathcal{L}}[A]))^{t-x} (\frac{x}{t} - (p_{AX} u_{\mathcal{L}}[A]))^2\end{aligned} \tag{4.8}$$

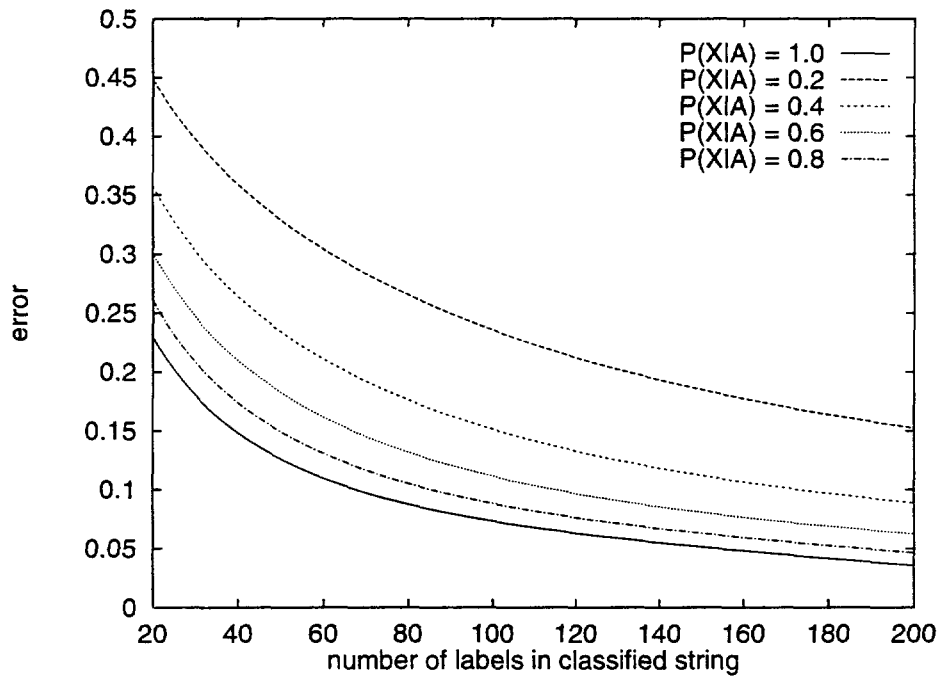

Figure 4.3: Classification error as a function of time (in terms of segments) for various values of $P(X|A) = p_{AX}$.

A plot of this equation in Fig. 4.3 shows how different values of p_{AX} can affect the convergence rate of the classification error. The lower the probability of correctly recognizing a sequence, the lower the effective occurrence frequency of the sequence is. Even though a sequence may be important to distinguish labeled languages, it may be negligible during language classification if the alignment process is not capable of recognizing the sequence.

4.2.2 Modeling Correlated Features

It is generally not true, however, that features are independent of each other. The confusion between features is usually such that they greatly influence each other's occurrence frequencies. Consider the confusion matrix of Table 4.3 which stands in contrast to Table 4.2, in which the features remain independent after alignment. In order to study the effect of such dependencies which may be frequent for sequences of shorter length, we formulate the problem in terms of two interdependent labels x and y.

Table 4.3: Confusion matrix due to the alignment with interdependent recognition of labels.

	aligned x	aligned y	aligned z
true a	0.8	0.1	0.1
true b	0.1	0.5	0.4
true c	0.0	0.8	0.2

We now proceed to compute the distribution of a label x in the aligned string based on the known distribution of labels a and b in the labeled string and a confusion matrix P denoting the probability of misrecognition due to the channel or alignment process depicted in Figure 4.2.

$$\begin{bmatrix} p_{ax} & p_{ay} \\ p_{bx} & p_{by} \end{bmatrix}$$

The distribution of the aligned features depends on two random processes: the distribution of the true data and the perturbation due to the alignment process. In order to calculate the distribution of label x, the random process of the alignment is first presumed to be constant; the result is then integrated taking the alignment into account. Given the definitions from Table 4.1, the expected value of n_x is calculated for a given n_a and n_b. Let ϕ^i_{ax} be a binary random variable which equals 1 when the ith occurrence of label a in the labeled string goes to x. Then, for a fixed number of labels in the labeled string, n_a and n_b, the expected variance of n_x can be calculated as follows. We first calculate the

expected value of n_x:

$$\begin{aligned} E[n_x|n_a n_b] &= E[\sum_{i=1}^{n_a} \phi_{ax}^i + \sum_{j=1}^{n_b} \phi_{bx}^j] \\ &= n_a E[\phi_{ax}^i] + n_b E[\phi_{bx}^j] \\ &= n_a p_{ax} + n_b p_{bx} \end{aligned} \tag{4.9}$$

In order to calculate the expected value of n_x by taking the distribution of a and b into account, $E[n_x|n_a n_b]$ is integrated over all n_a and n_b resulting in the following:

$$\begin{aligned} \Rightarrow E[n_x] &= \sum_{n_a} \sum_{n_b} E[n_x|n_a n_b] p(n_a n_b) \\ \mu_x &= \mu_a p_{ax} + \mu_b p_{bx} \end{aligned} \tag{4.10}$$

The expected value of n_x^2 is calculated in a similar manner by assuming a given n_a and n_b and then integrating. This results in the following derivation:

$$\begin{aligned} E[n_x^2|n_a n_b] &= E[(\sum_{i=1}^{n_a} \phi_{ax}^i + \sum_{j=1}^{n_b} \phi_{bx}^j)^2] \\ &= E[\sum_{i=1}^{n_a} \sum_{i'=1}^{n_a} \phi_{ax}^i \phi_{ax}^{i'} + 2 \sum_{i=1}^{n_a} \sum_{j=1}^{n_b} \phi_{ax}^i \phi_{bx}^j + \sum_{j=1}^{n_b} \sum_{j'=1}^{n_b} \phi_{bx}^j \phi_{bx}^{j'}] \end{aligned} \tag{4.11}$$

ϕ_{ax}^i and $\phi_{ax}^{i'}$ are not independent when $i = i'$, so that $E[\phi_{ax}^i \phi_{ax}^{i'}]$ is not equal to $E[\phi_{ax}^i]\ E[\phi_{ax}^{i'}]$. In fact, $\phi_{ax}^i \phi_{ax}^{i'}$ is either $1 * 1$ or $0 * 0$ and in both cases is the same as ϕ_{ax}^i. This explains the compensation summand which is shown in the next line of the derivation.

$$
\begin{aligned}
E[n_x^2|n_a n_b] &= \sum_{i=1}^{n_a}\sum_{i'=1}^{n_a} E[\phi_{ax}^i]E[\phi_{ax}^{i'}] + \sum_{i=1}^{n_a}(E[\phi_{ax}^i] - E[\phi_{ax}^i]^2) \\
&\quad + 2\sum_{i=1}^{n_a}\sum_{j=1}^{n_b} E[\phi_{ax}^i]E[\phi_{bx}^j] \\
&\quad + \sum_{j=1}^{n_b}\sum_{j'=1}^{n_b} E[\phi_{bx}^j]E[\phi_{bx}^{j'}] + \sum_{j=1}^{n_b}(E[\phi_{bx}^j] - E[\phi_{bx}^j]^2) \\
&= (n_a p_{ax} + n_b p_{bx})^2 + n_a p_{ax}(1 - p_{ax}) + n_b p_{bx}(1 - p_{bx})
\end{aligned}
\tag{4.12}
$$

Integrating over all values of n_a and n_b,

$$
\begin{aligned}
\Rightarrow E[n_x^2] &= \sum_{n_a}\sum_{n_b} E[n_x^2|n_a n_b]p(n_a n_b) \\
&= \sum_{n_a}\sum_{n_b}(n_a p_{ax} + n_b p_{bx})^2 + n_a p_{ax}(1 - p_{ax}) + n_b p_{bx}(1 - p_{bx})p(n_a)p(n_b) \\
&= (\mu_a^2 + \sigma_a^2)p_{ax}^2 + (\mu_b^2 + \sigma_b^2)p_{bx}^2 \qquad \text{(by definition of variance)} \\
&\quad + 2\mu_a\mu_b p_{ax}p_{bx} + \mu_a p_{ax}(1 - p_{ax}) + \mu_b p_{bx}(1 - p_{bx}) \\
&= (\mu_a p_{ax} + \mu_b p_{bx})^2 + \sigma_a^2 p_{ax}^2 + \sigma_b^2 p_{bx}^2 \\
&\quad + \mu_a p_{ax}(1 - p_{ax}) + \mu_b p_{bx}(1 - p_{bx})
\end{aligned}
\tag{4.13}
$$

We can now calculate the variance of x, since $\sigma_{n_x}^2 = E[n_x^2] - E[n_x]^2$, resulting in the following:

$$
\sigma_x^2 = \sigma_a^2 p_{ax}^2 + \sigma_b^2 p_{bx}^2 + \mu_a p_{ax}(1 - p_{ax}) + \mu_b p_{bx}(1 - p_{bx}) \tag{4.14}
$$

Using the same notation as above we can also calculate the covariance for distributions of x and y. The result will be needed in the next section but it is convenient to perform the derivation here.

$$
\begin{aligned}
E[n_x n_y | n_a n_b] &= E[(\sum_{i=1}^{n_a} \phi_{ax}^i + \sum_{i=1}^{n_b} \phi_{bx}^j)(\sum_{i=1}^{n_a} \phi_{ay}^i + \sum_{i=1}^{n_b} \phi_{by}^j)] \\
&= E[\sum_{i=1}^{n_a} \sum_{i'=1}^{n_a} \phi_{ax}^i \phi_{ay}^{i'}] \\
&\quad + E[\sum_{i=1}^{n_a} \sum_{j=1}^{n_b} \phi_{ax}^i \phi_{by}^j] \\
&\quad + E[\sum_{i=1}^{n_a} \sum_{j=1}^{n_b} \phi_{ay}^i \phi_{bx}^j] \\
&\quad + E[\sum_{j=1}^{n_b} \sum_{j'=1}^{n_b} \phi_{bx}^j \phi_{by}^{j'}]
\end{aligned}
\tag{4.15}
$$

ϕ_{ax}^i and $\phi_{ay}^{i'}$ are not independent when $i = i'$. ϕ_{ax}^i is always opposite to $\phi_{ay}^{i'}$ because for the given label i only one of the two cases can be true: either the ith label a goes to y or to x. Hence, $\phi_{ax}^i \phi_{ay}^{i'}$ is always zero for $i = i'$. Since $E[\phi_{ax}^i \phi_{ay}^{i'}]$ is not equal to $E[\phi_{ax}^i]E[\phi_{ay}^{i'}]$ a compensation summand is added as shown in the next line of the derivation.

$$
\begin{aligned}
&= \sum_{i=1}^{n_a} \sum_{i'=1}^{n_a} E[\phi_{ax}^i]E[\phi_{ay}^{i'}] - \sum_{i=1}^{n_a} E[\phi_{ax}^i]E[\phi_{ay}^i] \\
&\quad + \sum_{i=1}^{n_a} \sum_{j=1}^{n_b} E[\phi_{ax}^i]E[\phi_{by}^j] \\
&\quad + \sum_{i=1}^{n_a} \sum_{j=1}^{n_b} E[\phi_{ay}^i]E[\phi_{bx}^j] \\
&\quad + \sum_{j=1}^{n_b} \sum_{j'=1}^{n_b} E[\phi_{bx}^j]E[\phi_{by}^{j'}] - \sum_{j=1}^{n_b} E[\phi_{bx}^j]E[\phi_{by}^j] \\
&= n_a^2 p_{ax} p_{ay} + n_b^2 p_{bx} p_{by} + n_a n_b p_{ax} p_{by} + n_a n_b p_{bx} p_{ay} \\
&\quad - n_a p_{ax} p_{ay} - n_b p_{bx} p_{by}
\end{aligned}
\tag{4.16}
$$

Integrating over the distributions of a and b we get:

$$
\begin{aligned}
\Rightarrow E[n_x n_y] &= (\mu_a p_{ax} + \mu_b p_{bx})(\mu_a p_{ay} + \mu_b p_{by}) \\
&\quad - \mu_a p_{ax} p_{ay} - \mu_b p_{bx} p_{by} \\
&\quad + \sigma_a^2 p_{ax} p_{ay} + \sigma_b^2 p_{bx} p_{by} \\
&= \mu_x \mu_y - \mu_a p_{ax} p_{ay} - \mu_b p_{bx} p_{by} + \sigma_a^2 p_{ax} p_{ay} + \sigma_b^2 p_{bx} p_{by}
\end{aligned} \tag{4.17}
$$

We have now derived the distributions of the labels x and y in the data after alignment based on the distributions of a and b in the true data in conjunctions with the channel characteristics given in the confusion matrix. It has also been shown that the covariance of the labels after alignment is no longer zero.

4.2.3 Language Discrimination

As was done for true data in Section 4.1 we want to estimate the error for the data after alignment. In order to estimate the discrimination error due to the language-dependent output distributions of the classifier o (as shown in Figure 4.2), we derive the two parameters describing the distributions, the mean μ_o and the variance σ_o. Unlike Section 4.1, o now depends on x and y which in turn depend on a and b and P, the confusion matrix defining the misrecognition due to the alignment. In this section we show how the alignment process has affected the language discriminability of label a in the true data, corresponding to label x after alignment. For this purpose there is only one feature and we assume that the weighting of this feature $\alpha_x = 1$. We get:

$$
\begin{aligned}
E[n_o] &= E[n_x] \\
\Rightarrow \mu_o &= \mu_x \\
&= u_a p_{ax} + u_b p_{bx} && \text{(From Eq. 4.10)} \\
E[n_o^2] - E[n_o]^2 &= E[n_x^2] - E[n_x]^2 \\
\Rightarrow \sigma_o^2 &= \sigma_x^2 \\
&= \sigma_a^2 p_{ax}^2 + \sigma_b^2 p_{bx}^2 \\
&\quad + \mu_a p_{ax}(1 - p_{ax}) + \mu_b p_{bx}(1 - p_{bx}) && \text{(From Eq. 4.14)}
\end{aligned} \tag{4.18}
$$

Table 4.4: Specifications for model language τ which has one language discriminating feature.

$u_{\mathcal{L}_1}[a]$	$= .03;$	$u_{\mathcal{L}_2}[a]$	$= .01;$
$u_{\mathcal{L}_1}[b]$	$= .01;$	$u_{\mathcal{L}_2}[b]$	$= .01;$
$s_{\mathcal{L}_1}[a]$	$= .01;$	$s_{\mathcal{L}_2}[a]$	$= .01;$
$s_{\mathcal{L}_1}[b]$	$= .01;$	$s_{\mathcal{L}_2}[b]$	$= .01;$

(4.19)

In order to gain some intuitive understanding of the effect of the alignment on the discrimination error a model language with the specifications in Table 4.2.3 was implemented.

Note that there is only one discriminating feature. The confusion matrix P is given as follows.

$$\begin{bmatrix} p_{ax} & (1 - p_{ax}) \\ p_{bx} & (1 - p_{bx}) \end{bmatrix}$$

p_{ax} and p_{bx} are varied between 0 and 1 to study the effect of the alignment process on the distribution of label x. Figure 4.4 plots the effect of the alignment on the discrimination error as a function of p_{ax} and p_{bx}. As expected, the error is at a minimum when the occurrence frequency of the aligned features is undiluted by the alignment process, ie. $p_{ax} = 1$ and $p_{bx} = 0$. While $p_{ax} = 1$ and p_{bx} increases to one, the discrimination error increases as well, even though the distribution of label b is neutral. Alignment errors of any form increase the variance of x in this case and hurt discriminability. Note that error depends on the difference between the means of the occurrence frequencies of labels a and b in the true data distribution. This difference is 0 for a and .2 for b according to the specifications, which explains the asymmetry in the graph.

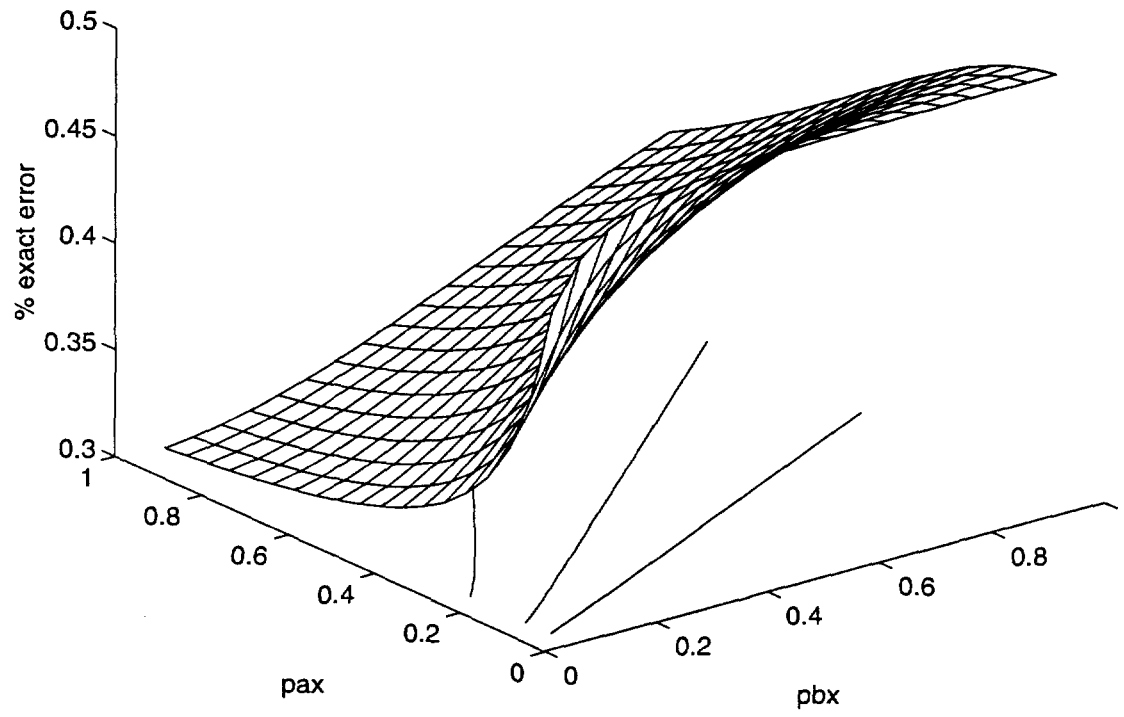

Figure 4.4: Discrimination error using exact sequence matching of x as a function of p_{ax} and p_{bx}.

4.3 Modeling Reconstructed True Data

In the previous section it was shown that the alignment can increase the discrimination error between two languages. The question to be answered in this section is whether the degradation in the language discrimination can be reversed. In other words, is it possible to use the knowledge of the transmission characteristics to reestimate the original, true distribution of the features? The goal then is to recompute the distribution of label a, given the distribution of labels x and y in the data after the alignment process due to the confusion matrix P. Let A be the input feature vector to the channel and X the output feature vector after the alignment. Then, A and X depend on P as follows:

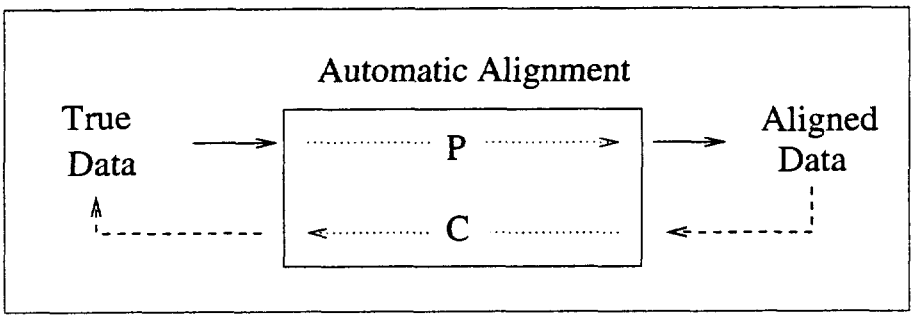

Figure 4.5: Alignment process and reverse process characterictics.

$$
\begin{aligned}
X &= P^T A \\
A &= P^{T^{-1}} X \\
A &= CX
\end{aligned}
\tag{4.20}
$$

We now want to compute C, the inverse of matrix P^T, to capture the inverse process due to the alignment as depicted in Figure 4.5. Let C be denoted by:

$$
\begin{bmatrix} c_{xa} & c_{ya} \\ c_{xb} & c_{yb} \end{bmatrix}
$$

4.3.1 Feature Modeling

The goal is to reestimate the distribution of a in the true data from the distributions of x and y in the aligned data. The reestimated value is denoted by $\tilde{a}$. We first derive $E[n_{\tilde{a}}^2]$ and $E[n_{\tilde{a}}]^2$.

$$
\begin{aligned}
E[n_{\tilde{a}}|n_x n_y] &= E[\textstyle\sum_{i=1}^{n_x} c_{xa} + \sum_{i=1}^{n_y} c_{ya}] \\
&= E[n_x c_{xa} + n_y c_{ya}] \\
\Rightarrow E[n_{\tilde{a}}] &= E[n_x]c_{xa} + E[n_y]c_{ya}
\end{aligned}
\tag{4.21}
$$

We know that:

$$
\begin{aligned}
E[n_x] &= \mu_a \alpha_{ax} + \mu_b \alpha_{bx} && \text{from Equation 4.10} \\
&= \mu_x \\
E[n_y] &= \mu_a \alpha_{ay} + \mu_b \alpha_{by} && \text{from Equation 4.10} \\
&= \mu_y
\end{aligned}
\tag{4.22}
$$

Integrating Equation 4.21 over distributions for a and b, we get:

$$
\begin{aligned}
\Rightarrow \mu_{\tilde{a}} &= \mu_x c_{xa} + \mu_y c_{ya} \\
E[n_{\tilde{a}}]^2 &= (\mu_x c_{xa} + \mu_y c_{ya})^2
\end{aligned}
\tag{4.23}
$$

$E[n_{\tilde{a}}^2]$ is calculated in a similar manner:

$$
\begin{aligned}
E[n_{\tilde{a}}^2 | n_x n_y] &= E[(\sum_{i=1}^{n_x} c_{xa} + \sum_{i=1}^{n_y} c_{ya})^2] \\
&= E[\sum_{i=1}^{n_x} \sum_{i'=1}^{n_x} c_{xa} + 2 \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} c_{xa} c_{ya} + \sum_{j=1}^{n_y} \sum_{j'=1}^{n_y} c_{ya}] \\
&= E[n_x^2 c_{xa}^2 + 2 n_x n_y c_{xa} c_{ya} + n_y^2 c_{ya}^2] \\
&= E[n_x^2] c_{xa}^2 + 2 E[n_x n_y] c_{xa} c_{ya} + E[n_y^2] c_{ya}^2
\end{aligned}
\tag{4.24}
$$

We know that

$$
\begin{aligned}
E[n_x^2] &= \sigma_x^2 + \mu_x^2 && \text{by def. of variance} \\
E[n_y^2] &= \sigma_y^2 + \mu_y^2 && \text{by def. of variance} \\
E[n_y n_x] &= \mu_x \mu_y \\
&\quad - \mu_a \alpha_{ax} \alpha_{ay} - \mu_b \alpha_{bx} \alpha_{by} \\
&\quad + \sigma_a^2 p_{ax} p_{ay} + \sigma_b^2 p_{bx} p_{by} && \text{(by Eq. 4.17)}
\end{aligned}
\tag{4.25}
$$

Equation 4.24 then becomes

$$
\begin{aligned}
E[n_{\tilde{a}}^2] = &(\mu_x c_{xa} + \mu_y c_{ya})^2 \\
&+ \sigma_x^2 c_{xa}^2 + \sigma_y^2 c_{ya}^2 \\
&- 2c_{ya}c_{xa}(\mu_a p_{ax} p_{ay} + \mu_b p_{bx} p_{by} \\
&+ 2c_{ya}c_{xa}\sigma_a^2 p_{ax} p_{ay} + \sigma_b^2 p_{bx} p_{by})
\end{aligned} \tag{4.26}
$$

so that

$$
\begin{aligned}
\Rightarrow \sigma_a^2 &= E[n_a^2] - E[n_a]^2 \\
&= \sigma_x^2 c_{xa}^2 + \sigma_y^2 c_{ya}^2 \\
&\quad - 2c_{ya}c_{xa}(\mu_a p_{ax} p_{ay} + \mu_b p_{bx} p_{by}) \\
&\quad + 2c_{ya}c_{xa}(\sigma_a^2 p_{ax} p_{ay} + \sigma_b^2 p_{bx} p_{by})
\end{aligned} \tag{4.27}
$$

In addition we will calculate the covariance of n_a and n_b which will become useful during the derivation of the output distributions of the classifier but is convenient to derive here.

$$
\begin{aligned}
E[n_{\tilde{a}} n_{\tilde{b}} | n_x n_y] &= E[(\textstyle\sum_{i=1}^{n_x} c_{xa} + \sum_{i=1}^{n_y} c_{ya})(\sum_{i=1}^{n_x} c_{xb} + \sum_{i=1}^{n_y} c_{yb})] \\
\Rightarrow E[n_{\tilde{a}} n_{\tilde{b}}] &= E[n_x^2] c_{xa} c_{xb} + E[n_y^2] c_{ya} c_{yb} \\
&\quad + E[n_x n_y] c_{xa} c_{yb} + E[n_x n_y] c_{ya} c_{xb} \\
&= (\mu_x^2 + \sigma_x^2) c_{xa} c_{xb} + (\mu_y^2 + \sigma_y^2) c_{ya} c_{yb} \\
&\quad + E[n_x n_y] c_{xa} c_{yb} + E[n_x n_y] c_{ya} c_{xb}
\end{aligned} \tag{4.28}
$$

where $E[n_x n_y]$ was calculated in Equation 4.17 and μ_x and σ_x are calculated in Equations 4.10 and 4.14. The covariance between $\tilde{a}$ and $\tilde{b}$ is non-zero.

4.3.2 Discrimination with Inexact Sequences

In order to estimate the discrimination error due to the language-dependent distributions of the reestimated $\tilde{a}$, we now derive the mean $\mu_{\tilde{a}}$ and the variance $\sigma^2_{\tilde{a}}$ for each of the two distributions. Unlike Section 4.1 and Section 4.2 classification now depends on $\tilde{a}$ which in turn depends on the inverse characteristic of the recognition process, modeled by C and on x and y which in turn depend on a and b and P the confusion matrix as shown in Figure 4.5. The variance of o is now due to the weighted sum of the reestimated distributions $\tilde{a}$ and $\tilde{b}$.

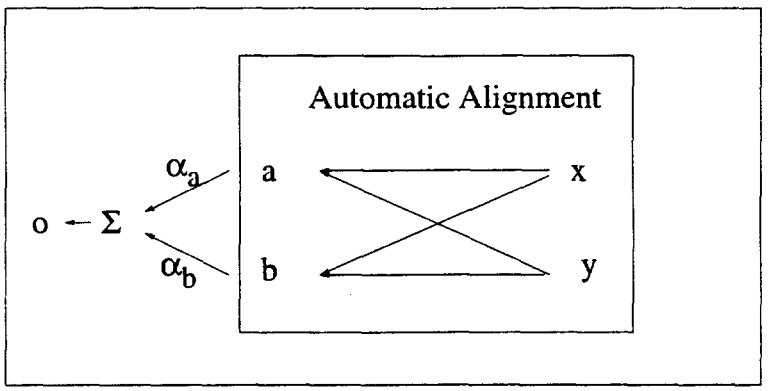

Figure 4.6: Classifier based on reestimated true data.

$$
\begin{aligned}
E[n_o|n_{\tilde{a}}n_{\tilde{b}}] &= E[\sum_{i=1}^{n_{\tilde{a}}} \alpha_{\tilde{a}} + \sum_{i=1}^{n_{\tilde{b}}} \alpha_{\tilde{b}}] \\
&= E[n_{\tilde{a}}]\alpha_{\tilde{a}} + E[n_{\tilde{b}}]\alpha_{\tilde{b}} \\
\Rightarrow \mu_o &= \mu_{\tilde{a}}\alpha_{\tilde{a}} + \mu_{\tilde{b}}\alpha_{\tilde{b}} \\
E[n_o]^2 &= (\mu_{\tilde{a}}\alpha_{\tilde{a}} + \mu_{\tilde{b}}\alpha_{\tilde{b}})^2 \\
&= \mu^2_{\tilde{a}}\alpha^2_{\tilde{a}} + \mu^2_{\tilde{b}}\alpha^2_{\tilde{b}} + 2\mu_{\tilde{a}}\mu_{\tilde{b}}\alpha_{\tilde{a}}\alpha_{\tilde{b}}
\end{aligned}
\tag{4.29}
$$

Similarly,

$$
\begin{aligned}
E[n_o^2|n_{\tilde{a}}n_{\tilde{b}}] &= E[(\sum_{i=1}^{n_{\tilde{a}}}\alpha_{\tilde{a}} + \sum_{i=1}^{n_{\tilde{b}}}\alpha_{\tilde{b}})^2] \\
&= E[n_{\tilde{a}}^2]\alpha_{\tilde{a}}^2 + E[n_{\tilde{b}}^2]\alpha_{\tilde{b}}^2 + 2E[n_{\tilde{a}}n_{\tilde{b}}]\alpha_{\tilde{a}}\alpha_{\tilde{b}} \\
\Rightarrow E[n_o^2] &= \mu_{\tilde{a}}^2\alpha_{\tilde{a}}^2 + \mu_{\tilde{b}}^2\alpha_{\tilde{b}}^2 + \sigma_{\tilde{a}}^2\alpha_{\tilde{a}}^2 + \sigma_{\tilde{b}}^2\alpha_{\tilde{b}}^2 + 2E[n_{\tilde{a}}n_{\tilde{b}}]\alpha_{\tilde{a}}\alpha_{\tilde{b}}
\end{aligned}
\tag{4.30}
$$

Finally, we get the variance as follows:

$$
\begin{aligned}
\sigma_o^2 &= E[n_o^2] - E[n_o]^2 \\
\Rightarrow \sigma_o^2 &= \sigma_{\tilde{a}}^2\alpha_{\tilde{a}}^2 + \sigma_{\tilde{b}}^2\alpha_{\tilde{b}}^2 + 2\alpha_{\tilde{a}}\alpha_{\tilde{b}}(E[n_{\tilde{a}}n_{\tilde{b}}] - E[n_{\tilde{a}}]E[n_{\tilde{b}}])
\end{aligned}
\tag{4.31}
$$

$E[n_{\tilde{a}}n_{\tilde{b}}]$ was calculated in Equation 4.28 and $E[n_{\tilde{a}}] = \mu_{\tilde{a}}$ and $E[n_{\tilde{b}}] = \mu_{\tilde{b}}$.

4.4 Classifying Model Language τ

Before comparing two systems using exact and inexact sequence matching, we would like to review why inexact sequence matching is believed to help the performance of a language identification system. The goal of inexact sequence matching is to associate sequences which cover (a) the variability of a word within one language without sacrificing any discriminability across languages and (b) model better the statistics of long strings even with limited training data. Thus, it is believed that combining sequences in this manner will result in a system which is more robust and therefore generalizes better to previously unseen test data. As an example consider the word "and" in English which, in fluent speech, is pronounced in several different ways: (**/a/n/dcl/d/**,**/n/dcl/**,**/a/n/**,etc.). These sequences of phonemes can then be associated with each other to form a single feature which intuitively would be more robust across different speakers. In this section we will analyze the effect of associating such sequences as opposed to matching only the most frequent sequence among them, for example **/a/n/dcl/d/**.

4.4.1 Exact vs. Inexact Sequence Matching

A sequence is chosen as a feature because it is language specific. In other words it occurs in language 1 with higher frequency than it does in language 2. This is the only type of feature of interest. The question now is: How can inexact sequence matching be useful to create a new feature distribution which will improve language discrimination? Intuitively, one might expect to add another sequence which similarly occurs more frequently in language 1 than in language 2. Returning to our example above, we can choose to add the sequences **/n/dcl/d/** to **/a/n/dcl/d/**, representing the English word for "and", and treat them as two instances of the same feature with inexact sequence matching. In this case suppose that both sequences are typical for English but not for German. This is the first case that will be considered in a theoretical analysis. However, to study the complete set of possibilities, we will present two more cases, the second one, in which the feature that is added can by itself not be used to discriminate between the languages, and the third one, in which the feature that is added belongs to another language. The three cases

that are considered are enumerated below.

1. Case 1 Adding a sequence b common in the same language as sequence a.

2. Case 2 Adding a neutral sequence b to a.

3. Case 3 Adding a sequence b common in the language where a is less common.

In each of the above cases, sequences from the aligned data are combined to form the inexact match in order to reestimate the input distribution of a as depicted in Figure 4.6. In all specifications, the distribution of the feature to be matched exactly occurs with higher frequency in the true data for language 1, making it a useful feature for discrimination. In order to see how the inexact sequence matching relates to exact sequence matching we will compare the discrimination error due to the following two distributions:

Distribution of Classifier Output Using Inexact Sequence Matching ($\tilde{a}$):

$$\begin{aligned}
\mu_o &= \mu_x c_{xa} + \mu_y c_{ya} && \text{(from Eq. 4.27)} \\
\sigma_o^2 &= \sigma_x^2 c_{xa}^2 + \sigma_y^2 c_{ya}^2 && \text{(from Eq. 4.23)} \\
&\quad -2c_{ya}c_{xa}(\mu_a p_{ax} p_{ay} + \mu_b p_{bx} p_{by}) \\
&\quad +2c_{ya}c_{xa}(\sigma_a^2 p_{ax} p_{ay} + \sigma_b^2 p_{bx} p_{by})
\end{aligned} \tag{4.32}$$

Distribution of Classifier Output Using Exact Sequence Matching (x):

$$\begin{aligned}
\mu_o &= u_a p_{ax} + u_b p_{bx} && \text{(From Eq. 4.10)} \\
\sigma_o^2 &= \sigma_a^2 p_{ax}^2 + \sigma_b^2 p_{bx}^2 \\
&\quad +\mu_a p_{ax}(1 - p_{ax}) + \mu_b p_{bx}(1 - p_{bx}) && \text{(From Eq. 4.14)}
\end{aligned} \tag{4.33}$$

We will see that for all cases where inexact sequence matching outperforms exact sequence matching there exists a way of achieving higher classification by using exact sequence matching in a different way. The confusion matrix for the theoretical analysis is given by Equation 4.34.

$$\begin{bmatrix} p_{ax} & (1-p_{ax}) \\ p_{bx} & (1-p_{bx}) \end{bmatrix} \tag{4.34}$$

In order to generalize to a larger number of inexact sequences to be grouped, the confusion matrix is also used for generating the model language in some cases as discussed below. The confusion matrix P in theory does not restrict specifications to substitutions but can include deletions and insertions. Without loss of generality, we simplify the implementation to include only substitutions.

$$\begin{bmatrix} p_{ax} & (1-p_{ax})/2 & (1-p_{ax})/2 \\ p_{bx} & (1-p_{bx}) & 0 \\ p_{cx} & 0 & (1-p_{cx}) \end{bmatrix} \tag{4.35}$$

4.4.2 Case 1: Adding Common Sequences

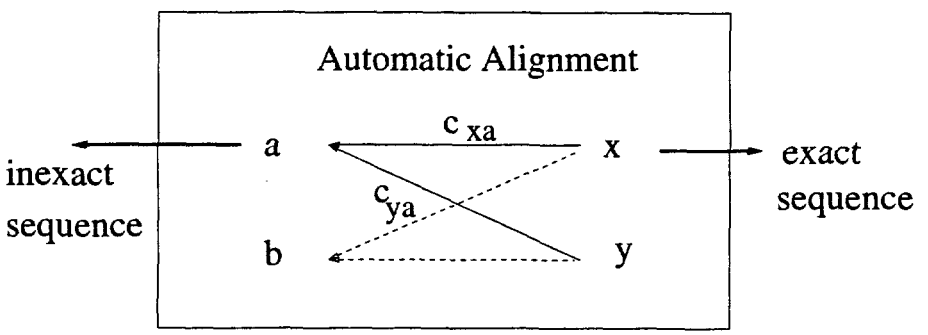

Figure 4.7: *Discrimination is better when using two separate features rather than reestimating* the distribution of $\tilde{a}$

In the first case to be analyzed, all features to be combined occur with higher frequency in language 1. The specifications for an example case are given in Table 4.5. p_{ax} and p_{bx} are varied between zero and one in order to calculate the discrimination error for $\tilde{a}$

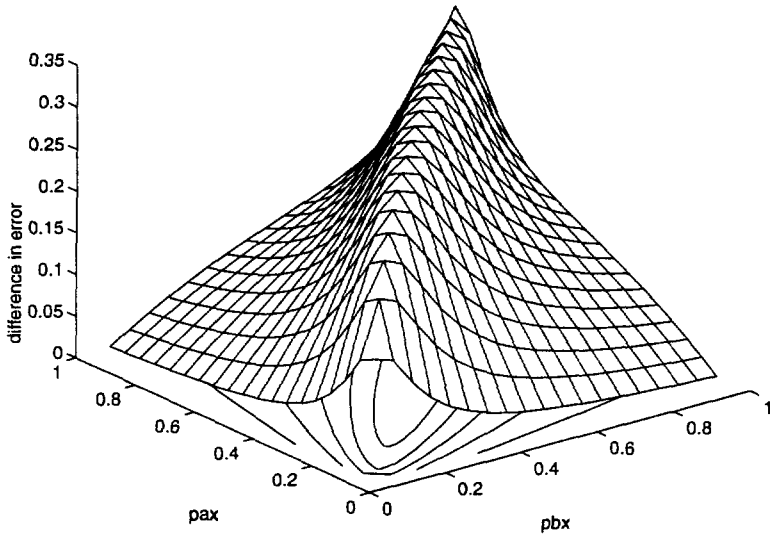

Figure 4.8: Surface plot for the difference in error for exact vs. inexact sequence matching using specifications given in Table. 4.5. The surface would lie below zero if inexact matching were to outperform exact sequence matching.

Table 4.5: Specification for case 1

$u_{\mathcal{L}_1}[a] = .03;$	$u_{\mathcal{L}_2}[a] = .01;$
$u_{\mathcal{L}_1}[b] = .03;$	$u_{\mathcal{L}_2}[b] = .01;$
$s_{\mathcal{L}_1}[a] = .01;$	$s_{\mathcal{L}_2}[a] = .01;$
$s_{\mathcal{L}_1}[b] = .01;$	$s_{\mathcal{L}_2}[b] = .01;$

(4.36)

using both sequences y and x for inexact sequence matching. In order to see if the error is reduced compared to using exact sequence matching, Figure 4.8 plots the difference in error between both approaches. Since it can be seen from the plot that the difference is never below zero, the error will always be larger for inexact sequence matching than for exact sequence matching. For this case, inexact sequence matching never outperforms exact sequence matching even though we expected to gain by grouping the sequences representing different pronunciations together; it is preferable to simply use exact sequence matching. as illustrated in Figure 4.7. An intuitive way of looking at this result is to realize that there are two useful features in the aligned files. The optimal way of combining the two features can be calculated exactly. The chances are very low that the optimal weighting

of the sequences, maybe their non-linear combination, corresponds to the vector $\vec{\alpha}$ used to reestimate the labeled sequence.

4.4.3 Case 2: Adding Neutral Sequences

As in the example from Section 4.2, the distributions of the model language for this case are given in Table 4.6.

Table 4.6: Specification for case 2

$u_{\mathcal{L}_1}[a]$	$= .03;$	$u_{\mathcal{L}_2}[a]$	$= .01;$
$u_{\mathcal{L}_1}[b]$	$= .01;$	$u_{\mathcal{L}_2}[b]$	$= .01;$
$s_{\mathcal{L}_1}[a]$	$= .01;$	$s_{\mathcal{L}_2}[a]$	$= .01;$
$s_{\mathcal{L}_1}[b]$	$= .01;$	$s_{\mathcal{L}_2}[b]$	$= .01;$

(4.37)

In the confusion matrix P, given by Table 4.34, p_{ax} and p_{bx} are varied between zero and one in order to calculate the discrimination error for $\tilde{a}$ using both sequences y and x during inexact sequence matching to reestimate the labeled files. In order to see if the error is reduced compared to exact sequence matching of x (as plotted in Figure 4.4), Figure 4.9 plots the difference in the error percentage using the two approaches. Inexact sequence matching outperforms exact sequence matching for areas in which the surface plot lies below zero.

Table 4.7: Specifications for model language τ which has one language discriminating feature.

$u_{\mathcal{L}_1}[a]$	$= .03;$	$u_{\mathcal{L}_2}[a]$	$= .01;$
$u_{\mathcal{L}_1}[b]$	$= .01;$	$u_{\mathcal{L}_2}[b]$	$= .01;$
$u_{\mathcal{L}_1}[c]$	$= .01;$	$u_{\mathcal{L}_2}[c]$	$= .01;$
$s_{\mathcal{L}_1}[a]$	$= .01;$	$s_{\mathcal{L}_2}[a]$	$= .01;$
$s_{\mathcal{L}_1}[b]$	$= .01;$	$s_{\mathcal{L}_2}[b]$	$= .01;$
$s_{\mathcal{L}_1}[c]$	$= .01;$	$s_{\mathcal{L}_2}[c]$	$= .01;$

(4.38)

From Figure 4.9 it can be seen that inexact sequence matching outperforms exact sequence matching if $p_{ax} < p_{bx}$. In order to verify the theoretical prediction on generated

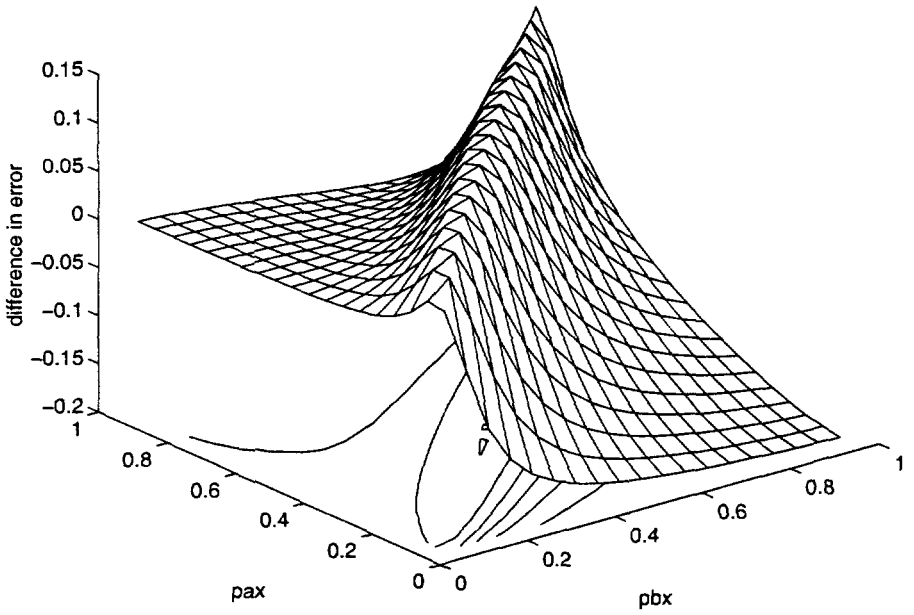

Figure 4.9: Surface plot for the difference in error for exact vs. inexact sequence matching using specifications given in Eq. 4.34. The surface lies below zero when inexact matching outperforms exact sequence matching ($p_{ax} < p_{bx}$).

data we choose an operating point at $p_{ax} = .1$ and $p_{bx} = p_{cx} = .7$ for which inexact sequence matching has higher discrimination than exact sequence matching. Generalizing to three sequences, the specifications are given in Equation 4.35 and Table 4.7. Figure 4.11 plots both the actual error and the theoretical prediction comparing exact and inexact sequence matching. It can be seen that inexact matching indeed outperforms exact sequence matching.

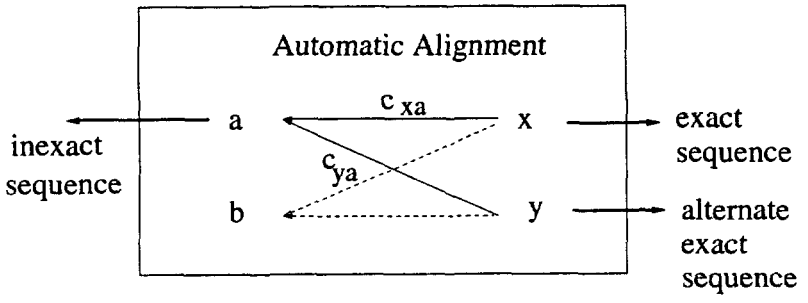

Figure 4.10: Discrimination is best when using the alternate feature y rather than x or reestimating the distribution of $\tilde{a}$

However, looking more closely at the resulting distributions in the aligned data, it can be seen that y has become a strong feature: Plugging $u_{\mathcal{L}_i}[j]$ and P into Equation 4.33 the difference in mean occurrence frequency of y in the two languages is .2 while the same difference for sequence x is only .04. Thus, the distributions for y for the languages are further apart indicating better discriminability. Figure 4.11 plots the discrimination error achieved by spotting sequence y exactly rather than performing inexact sequence matching on x, y, and z, trying to reestimate the original distribution of a. This shows that, even though there exists an operating point at which inexact sequence matching outperforms exact sequence matching, another sequence at this operating point was transformed into a feature whose exact matching discriminates better than the inexact sequence matching. Intuitively one may think of the discriminating information to be in either of the two sequences in the aligned files. Depending on the probability of recognition, the discriminant information is transfered from label x to y in Figure 4.10.

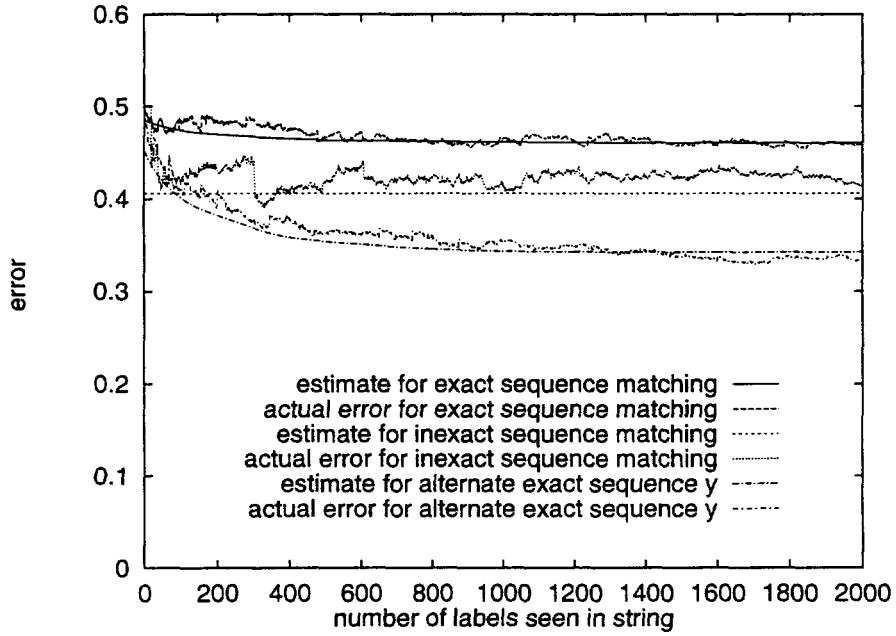

Figure 4.11: Plot of discrimination error for case 2. Exact sequence matching outperforms inexact sequence matching when the correct label is matched.

4.4.4 Case 3: Adding Opposing Sequences

In this case the distribution is constructed so that the second sequence, b, is a feature for language 2, while a is a feature for language 1. The specifications for the example are given in Table 4.8.

Table 4.8: Specification for case 3

$u_{\mathcal{L}_1}[a] = .03;$	$u_{\mathcal{L}_2}[a] = .01;$
$u_{\mathcal{L}_1}[b] = .01;$	$u_{\mathcal{L}_2}[b] = .03;$
$s_{\mathcal{L}_1}[a] = .01;$	$s_{\mathcal{L}_2}[a] = .01;$
$s_{\mathcal{L}_1}[b] = .01;$	$s_{\mathcal{L}_2}[b] = .01;$

(4.39)

Again, p_{ax} and p_{bx} are varied between zero and one in order to calculate the discrimination error for $\tilde{a}$ using both sequences y and x for inexact sequence matching to reestimate the distribution of a in the labeled files. In order to see if the error is reduced compared to using exact sequence matching, Figure 4.12 plots the difference in error for both approaches. Inexact sequence matching outperforms exact sequence matching for areas in which the surface plot lies below zero. It can be seen that inexact sequence matching is useful for almost all cases of inexact alignment in this example.

Table 4.9: Specifications for model language τ which has one language discriminating feature.

$u_{\mathcal{L}_1}[a] = .03;$	$u_{\mathcal{L}_2}[a] = .01;$
$u_{\mathcal{L}_1}[b] = .01;$	$u_{\mathcal{L}_2}[b] = .03;$
$u_{\mathcal{L}_1}[c] = .01;$	$u_{\mathcal{L}_2}[c] = .03;$
$s_{\mathcal{L}_1}[a] = .01;$	$s_{\mathcal{L}_2}[a] = .01;$
$s_{\mathcal{L}_1}[b] = .01;$	$s_{\mathcal{L}_2}[b] = .01;$
$s_{\mathcal{L}_1}[c] = .01;$	$s_{\mathcal{L}_2}[c] = .01;$

(4.40)

Generalizing to three sequences with the specifications, given in Tables 4.9 and 4.35, we choose a point of operation at $p_{ax} = .1$ and $p_{bx} = p_{cx} = .7$. such that inexact sequence matching has higher discrimination than exact sequence matching according to Figure 4.12. However, in the beginning of this section we stated that the motivation for

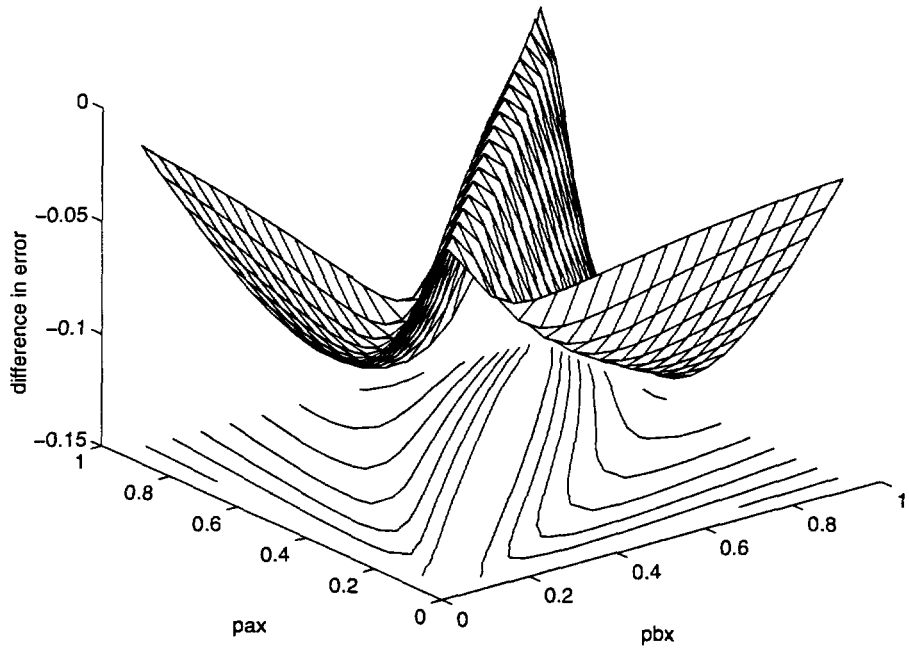

Figure 4.12: Surface plot for the difference in error for exact vs. inexact sequence matching using specifications given in Eq. 4.8. The surface lies below zero when inexact matching outperforms exact sequence matching.

inexact sequence matching was to associate sequences to cover the variability of a "word" within a language. In this particular case the two features, a and b represent opposing languages and there exists motivation for treating them separately. Thus, building a system that was already shown in Section 4.2 (see Figure 4.2), spotting two features exactly in the aligned data outperforms inexact sequence matching. Figure 4.13 verifies this by plotting the corresponding error curves from the specified model language.

In conclusion it can be seen that inexact sequence matching did not help in the examples that were presented. The same trends that were illustrated in this chapter, occur when the standard deviations and means from the specifications are varied. In general it seems that the reestimation of the features perturbs the data even more and allows no gain. In some cases it may therefore be preferable to increase the number of features rather than to use them in inexact sequence matching. Although we have not been able to show this analytically, our empirical analysis thus suggests that inexact sequence matching is

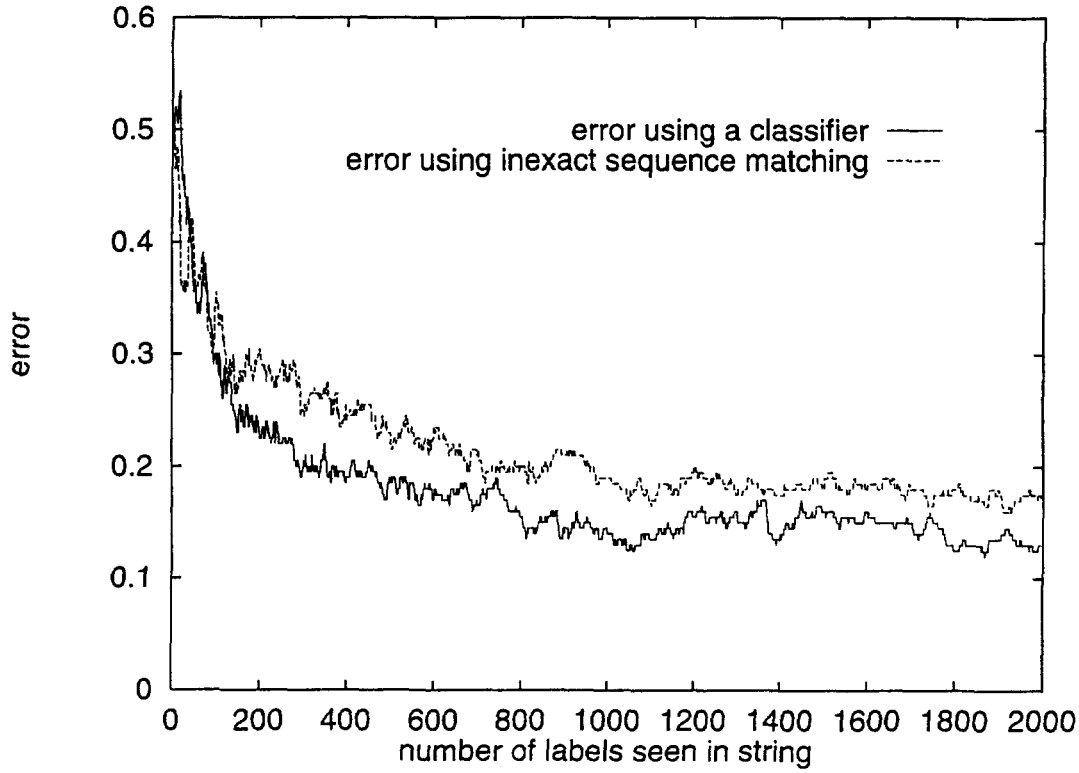

Figure 4.13: Plot of discrimination error for case 2. Treating features separately is preferable to combining features in inexact sequence matching.

less attractive than it appears. We will investigate this further with practical examples in Appendix E. As in Section 4.4.2, an intuitive way of looking at this result is to realize that there are two useful features in the aligned files. The first feature captures discriminant information about the first language and the second feature captures information about the second language. The optimal way of combining the two features can be calculated exactly. Again, the chances are very low that the optimal weighting of the sequences, corresponds to the vector $\vec{\alpha}$ used to reestimate the labeled sequence.

Chapter 5

Language Identification of Telephone Speech

The algorithm described in the previous chapter is now applied to discriminate between English and German telephone speech. Compared to regular speech, telephone speech is noisy, thereby increasing the difficulty level of discriminating between the languages. In order to recognize this type of noisy speech we rely on previous research in speech recognition. We choose to use a signal representation called PLP (perceptual linear prediction) which has been shown to work well on telephone speech [38]. Once the signal is thus represented, a neural network is used to classify the signal in terms of a linguistic speech unit such as a phoneme for example [89].

An important part of this chapter is the selection of the speech units which the neural network will discriminate. By starting with the union of all phonemes across six languages in the database we can approximate a universal representation. We take advantage of the tradeoff between precision and accuracy to cluster phonemes across languages without losing the discriminating information. For this purpose we have developed the necessary theory in the preceding chapters. The theory allows us to predict the discriminability for a given set of clustered phonemes without implementing a system at each step of clustering. A final set of clustered phonemes represents the maximum precision necessary in order to achieve the same degree of discrimination obtained before clustering. The goal is to achieve a higher accuracy of phoneme cluster recognition after retraining a neural network classifier at the reduced level of modeling precision.

Speech is automatically aligned by combining the output values of the neural network

classifier with a Viterbi search which seeks to find the best possible sequence of speech units to align with a given speech input. This process results in a discrete linguistic representation of the speech, containing the speech representation which was developed by clustering phonemes. This alignment is performed on all speakers in the training and test set of our database. Each language is assumed to have a characteristic set of features which consists of sequences of the phoneme clusters in the aligned speech. From the training set such features are extracted and will be used to classify the language of the test set speakers. The process of sequence selection has been theoretically developed in Chapter 3.

The language of a given speaker is identified by aligning the speech with the set of derived phoneme clusters, extracting sequences of phoneme clusters as features, and employing these to classify the speech. This final classifier is also implemented by using a non-linear neural network trained to discriminate between German and English using a feature set derived from the speakers in the training set. In this chapter we will show the set of features used to discriminate the two languages. Results indicate that by clustering phonemes across languages we do not lose discriminative information. We also demonstrate the effect of the inaccurate alignment on language identification results, which was indicated in theory in Chapter 4.

In this chapter, Section 5.1 will describe the telephone speech data. The tools which will form the basis on which we implement the language identification system are discussed in Section 5.2. The speech representation is developed in Section 5.3. Section 5.4 describes the implementation of the final language identification system and results are given in Section 5.5.

5.1 The Data

The database used in this study is the $OGI-TS$ database [74], which is a multi-language telephone speech database including speech from 21 languages. This database is publically available and has been used as the standard database by the National Institute of Standard and Technology (NIST) for annual progress evaluations of language identification systems.

Before analyzing the languages in the database and explaining the level of labeling, we describe the data collection process.

5.1.1 Data Collection

Telephone speech data is collected over analog and digital telephone lines. For the analog lines speech was collected using a Gradient Technology Desklab connected via a SCSI port to a Sun 4/110 workstation. Since November 1993, the majority of data has been collected using a 24 channel T1 digital line connected to three LINKON FC3000 Communication Boards. These devices were programmed to answer the telephone, play digitized files in each of the languages requesting the speech samples, and digitize the callers' response for a designated period of time. Speech was sampled at 8000 samples per second at 14 bit resolution. The recording protocol was designed to obtain topic specific speech as well as free speech on any subject. In this thesis we exclusively use the data in response to a one minute story of the caller's choice. The speakers werre given 10 seconds to organize their thoughts before recording in order to minimize the number of pauses and false starts.

5.1.2 Languages

The Phonetically Labeled Corpus consists of the following languages: English, Japanese, German, Spanish, Hindi, and Mandarin. These include about 70 labeled files in each of the languages. In this thesis we mainly use the two languages English and German. Please refer to Table 5.1 for the amount of data available for each language.

5.1.3 Corpus Statistics

The training set consisted of 51 and 46 stories in English and German respectively, while the independent development test sets consisted of 19 English and 20 German stories. All story files were labeled with worldbet [39], a new ASCII encoding of the International Phonetic Alphabet (IPA) including non-European languages (see Appendix A). In most cases the symbols consist of a concatenation of an IPA symbol with diacritics. For our purposes, all diacritics have been stripped off and some phonemes have been merged based on linguistic knowledge due to a lack of sufficient samples for training, as shown in

Table 5.1: Number of files exceeding various durations.

Number of files	Training Set		Test Set	
Time (seconds)	EN	GE	EN	GE
5	51	46	19	20
10	51	46	19	20
15	51	46	19	20
20	51	46	19	20
25	51	46	19	19
30	51	46	19	19
35	51	45	19	19
40	51	45	19	19
45	49	43	19	18

Table 5.2. Table 5.3 lists the set of preclustered phonemes. This resulted in a set of 95 phonemes with which to label the training set.

Table 5.2: Table of labels for unmerged label set of 95 phonemes

Number of Labels	Labels
95	& , &r , .pau , 2 , 3 , 3r , 4r , 9r , > , > Y , ? , ?* , @ , A, A: , C , D , E , Eax , I , Ix , K , N , S , T , U , V , ^ , a , aI , aU , ai , b , bc , cC , cCc , cCh , d , dZ , dc , d[, d[c , d(, drc , e , e: , ei , f , g , gc , h , hs , i , i: , j , k , kH , kc , kh , l , m , n , nj , o , o: , oU , oax , p , pc , ph , r(, rr , s , sr , tS , tc , t[, t[H , t[c , t[c: , th , trc , ts , tsR , tsc, tsr , tsrc , u , u: , v , w , x , y , yax , z

Table 5.1 shows the number of files present at each length given in seconds for which results are analyzed in this chapter. Appendix A and Appendix B show the labeling conventions for the database used in this experiment as well as some statistics on phoneme occurrences. In addition Tables 5.4 - 5.5 give word examples corresponding to the respective labels. Some of the words illustrate the belief that merging of phonemes across languages is viable. Looking at Table 5.5 many words chosen as examples of the labels in the columns are very similar. Even though a linguist may detect the difference in pronunciation of and American /*l*/ in "loss" vs. the German /*l*/ in "los", the question is whether an automatic phoneme-recognizer will be able to make such a distinction. If it cannot make the distinction, the two phonemes may be acoustically close enough to be merged

Table 5.3: Table of premerged labels within final set of 95 phonemes

> Y > i	?* ?c	E 8
I If	K G	b bH
bc bc:	d dr	dZ dR dZH
dc dZc	d[d[: d[H d[z	g gH
i ih	kc kc:	l L l: l(
m m:	n n: n[nr	nj ng
o 7	oU ow	p pH
pc pc:	r(r(H	rr r r+
s s:	sr	tS tsH
t[H t[s	t[c t[sc	th t tR tSH
trc trc:	tsc tSc	u uax
y Y y:	z Z	

and treated as a single speech unit.

Table 5.4: Labels with word examples

LABEL	German Word	English Word
&	in	character
&r	-	oredered
.pau	-	-
2	-	-
3	-	-
3r	-	research
4r	-	-
9r	-	research
>	elektronik	ordered
> Y	neu	Lloyd
?	-	-
?*	-	-
@	nähe	interaction
A	-	talk
A:	Firma	-
C	ich	-
D	-	the
E	amerika	reception
Eax	sehr	-
I	ich	recent
Ix	ich	recent
K	groß	-
N	streng	mixing
S	streng	should
T	-	thought
U	und	put
V	-	-
^	-	about
a	firma	-
aI	-	I
aU	Frau	out
ai	meine	-
b	bin	bin
bc	bin	bin
cC	-	-
cCc	-	-
cCh	-	-
d	denn	danger
dZ	Virginia	subject
dc	denn	danger
d[	-	-
d[c	-	-
d(	-	ordered
drc	-	-

Table 5.5: Labels with word examples

LABEL	German word	English word
e	-	-
e:	leben	-
ei	-	phase
f	firma	firm
g	gehen	go
gc	gehen	go
h	heiß	hot
hs	-	-
i	-	-
i:	Klima	research
j	Jahren	years
k	-	-
kH	-	-
kc	Amerika	America
kh	Amerika	America
l	lang	long
m	Mitte	middle
n	nun	now
nj	-	realizing
o	gewöhnt	-
o:	groß	-
oU	-	low
oax	vor	-
p	-	-
pc	purpur	purple
ph	purpur	purple
r(	-	-
rr	gerade	-
s	los	loss
sr	-	-
tS	Deutsch	much
tc	elektronik	electronic
t[	-	-
t[H	-	-
t[c	-	-
t[c:	-	-
th	elektronik	electronic
trc	-	-
ts	zum	-
tsR	-	-
tsc	-	-
tsr	-	-
tsrc	-	-
u	zu	-
u:	gut	juice
v	wir	of
w	-	way
x	-	-
y	natürlich	-
yax	hier	-
z	so	phase

5.2 Speech Recognition - Alignment

The speech-recognition system used in this study employs neural networks for phonemic recognition, followed by a Viterbi search which time aligns the speech (wave) files with the labels.

5.2.1 Neural Network based Phoneme Classification

A neural network is used to assign scores relating the probability of seeing a given phoneme or speech unit at the input. The neural network classifiers used here are fully-connected, feed-forward networks trained using back-propagation with conjugate gradient optimization [4] using a mean squared error criterion. Such a neural network has three layers. Each node in the first layer corresponds to an input feature derived from the waveform. An equal number of frames from each of the phoneme classes are sampled according to the labeled database. In this thesis, a frame is defined to be a 6ms window of the digitized speech. The acoustic input is represented with a seventh order Perceptual Linear Predictive (PLP) model [38], yielding 8 coefficients (including one for energy). This representation is a modification of linear predictive coding taking into account knowledge about the human hearing mechanism. For each sampled frame, 56 ($= 8 \times 7$) PLP coefficients within a 156 msec window, centered on the frame to be classified, are computed and serve as input to the phonetic classifier. The sampling intervals are shown in Figure 5.1 and have been derived from work done by Roginski [89] who experimented with different window sizes to find the optimal static representation of a frame to be classified. The objective is to provide substantial contextual information about the chosen frame to the network. The number of hidden nodes was derived experimentally for best performance. Each output node in the third layer corresponds to a phoneme or speech unit.

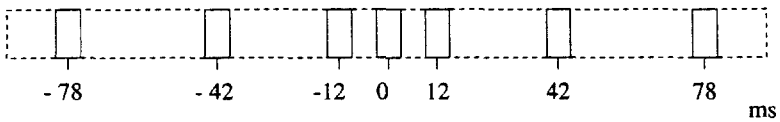

Figure 5.1: Sampling intervals for the PLP features. The solid boxes indicate the frames for which PLP coefficients are computed. Dashed boxes indicate skipped frames.

5.2.2 Aligning the Speech

Acoustic features as described above are calculated every 6 ms. Thus, the network assigns 95 phoneme category scores to each 6 ms time frame of the utterance, reflecting an estimate of the probability of seeing a phoneme at the input to the network. Figure 5.2 illustrates the alignment process, in which these output scores are computed for each incoming time frame creating a matrix of probability-like scores over time.

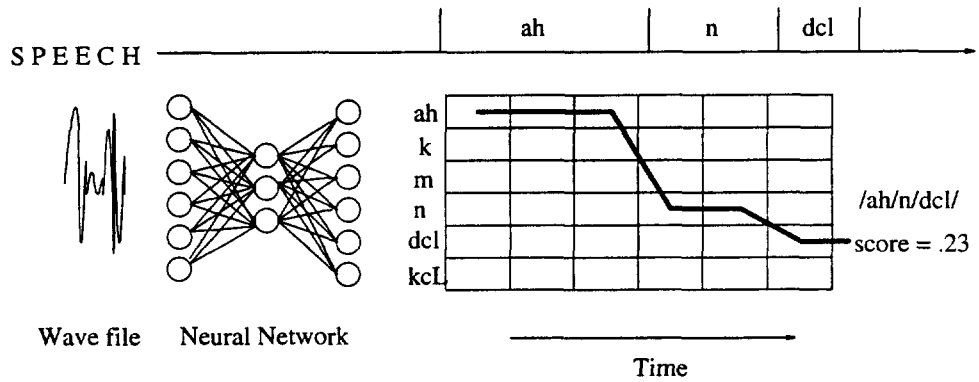

Figure 5.2: Automatic labeling of incoming utterance based on Viterbi search and neural network outputs.

Speech is segmented into a time-aligned string of phonemes by using the optimal path through the outputs of the neural network. Duration and transition probabilities are derived from the hand-labeled files. Durations are represented as minimum and maximum duration corresponding to the 2nd and 90th percentile of a histogram computed over all training files for each of the phonemes. A Viterbi search [1] takes duration constraints and transition probabilities into account when searching for the optimal path. Let the string of time-aligned phonemes consist of the sequence of phonemes $ph_j (j = 1, ..., J)$. If we assume that phoneme ph_j spans frames T_j to $T_{j+1} - 1$, the likelihood of phoneme ph_j with length T_j to $T_{j+1} - 1$ is then given by :

$$\prod_{T_j}^{T_{j+1}-1} p_t(ph_j) \tag{5.1}$$

[1]Thanks to Mark Fanty for explaining his algorithm which we used here.

The duration constraints are used by inforcing a penalty when the segments are shorter or longer than the given duration limits.

Shorter Segments: multiply score by $L*shortpenalty$, where L is the number of frames the segment is short of the lower limit.

Longer Segments: multiply score by $L*longpenalty$, where L is the number of frames the segment is above the upper limit.

Every time a transition is made from phoneme ph_{j-1} to ph_j, the score is multiplied by $P(ph_j|ph_{j-1})$, which may be scaled by a constant to achieve the correct balance between acoustic and language probabilities. Let $q_t(ph_j)$ be the score associated with a phoneme at time of transition including penalties, then the likelihood of the returned sequence of phonemes is given by:

$$q(ph_1) * \prod_{j=2}^{J} q(ph_j)P(ph_j|ph_{j-1}) \tag{5.2}$$

5.3 Speech Representation

As explained in Chapter 3, the clustering of phonemes across languages is based on the premise that not all phonemes are of equal importance to the language identification task. In this section, we will derive the phoneme clusters used for classifying English vs. German by clustering phonemes, estimating the discrimination error at each level, and pruning the clustering tree.

Clustering The clustering process is started by using the set of 95 phonemes as shown in Table 5.2. At each merger two phonemes are chosen which result in the maximal increase in the expected mutual information measure given by Equation 3.1.

$$Mutual\ \ Information = \sum_{x,y} p(y)p(x|y)log(\frac{p(y)p(x|y)}{p(x)p(y)})$$

In practice, $p(x|y)$ (the probability of seeing phoneme x after alignment given that the phoneme was labeled y) is an entry in the confusion matrix which is derived by aligning

utterances before clustering and comparing frame-based labels of the aligned files to the hand labeled files. Deriving the prior probability $p(y)$ from the labeled files, the mutual information between the observed and actual phonemes can now be calculated.

Merging of speech units is allowed while language classification does not fall below a threshold, set to the first estimated error before merging. This estimated error is based on statistics computed from the aligned files. Thus the distribution of sequence occurrence given by parameters u and s (Eq. 4.7) as defined in Chapter 3 can be estimated to include the probability of alignment. In order to know when a merge is disallowed, it is essential to have a good estimate of the language identification error. Assuming that the error is estimated correctly, we are able to prune the clustering tree such that the minimal error can be obtained.

Error Estimation The error is estimated for the list of chosen sequences, by representing their combined error as described in Chapter 3. Adding a word to a given list results in the following distributions for the language pair $i = 1,2$ (assuming independence of sequence occurrences):

$$List \simeq N(u_i[1] + \alpha u_i[2], s_i[1]^2 + \alpha^2 s_i[2]^2) = N(\mu_i, \sigma_i)$$

And their combined error is estimated as:

$$\frac{1}{2} e^{-\frac{1}{4}[\frac{(\mu_2-\mu_1)^2}{\sigma_1^2+\sigma_2^2}]}$$

In theory there is no limit to the number of sequences used for discriminating between two languages. However, in practice, as the number of sequences increases, the amount of training data becomes insufficient to estimate occurrence frequencies accurately. Thus, the risk of overtraining exists. In order to avoid this, cross validation is performed on two halves of the training set. The number of allowed sequences in the list is limited by setting a threshold based on the error estimation corresponding to each individual sequence.

Pruning Appendix C.1 shows the order of clustering, Appendix C.2 and Appendix C.3 show a detailed listing of allowed and disallowed merges, respectively. Using the expected mutual information measure for clustering, and language classification as criterion for

pruning the tree, 59 phoneme classes were determined as the optimal number of clusters. After the first encountered disallowed merge, an extra fourteen merges are gained. Examples of disallowed merges include: The American English flapped /**d**/ vs. the German rolled /**r**/. While acoustically close, they yield too much discriminating information to be merged. The English /**w**/ does not occur in German. Even though it is acoustically close to /**l**/, occurring in both languages, it is not merged because of its importance in discriminating the two languages. /**n**/ and /**ng**/, both occurring in "ending", are not merged: /**ng**/ is more frequent in English than in German as expected due to its grammatical function in the English participle. The merges just mentioned are those which would have contributed most to an error increase.

Results In order to evaluate the impact of clustering we compare frame-based results for the neural network and alignment accuracy before and after clustering. The neural networks trained to recognize 59 and 95 phonemes have 35 and 75 hidden nodes respectively. Results in Table 5.3 are reported in terms of frame based accuracy. The third column shows a 3% increase in accuracy for the merged phoneme set. The performance of the alignment for these 59 phoneme clusters compared to the previous result using 95 phonemes is given in the fourth column. It can be seen that there is an increase in correct alignment after merging phonemes. The column labelled "grammar" indicates the languages used to derive the grammar for alignment purposes. Before merging phonemes, the grammar that is used does not seem to have any effect on either frame-based or post-alignment recognition accuracy. However, the bigram probabilities derived from English and German do improve alignment accuracy at the customized level of 59 phoneme classes from 19% to 25%. This improvement in the alignment is expected to be reflected in the language identification accuracy.

5.4 Language Identification System

In this section we will describe the language identification system. In Section 5.3 we have developed a set of speech units which were used to align the English and German speech as described in Section 5.2. These aligned files are used for building a language identification

Table 5.6: Summary of Results from Alignment

Phoneme-classes	Grammar	Frame-based accuracy (%)	Alignment accuracy (%)
95	SIX	16	15
	ENGE	16	16
59	SIX	19	23
	ENGE	19	25

system in this section.

5.4.1 System Design

The implemented system will discriminate between English and German. This language pair was chosen because it is known in the community to be one of the most difficult pairs to discriminate within the phonemically labeled languages in the $OGI - TS$ database (a reflection of the linguistic similarities between these two strongly related languages). This chapter will describe a neural network implementation of the theoretical design developed previously; There are six essential steps in this process:

1) Language-dependent clustering of phonemes
2) Automatic alignment of utterances with phoneme-clusters
3) Building of a database which records occurring sequences
4) Feature selection
5) Sequence spotting
6) Language classification

Figure 5.3 depicts a flowchart with implementation details. Both training and test data are automatically aligned with the specified set of tokens. From two halves of the aligned training data, two databases of the sequences occurring in the respective training sets are built. From the databases, a set of sequences is chosen as features to identify the aligned test data. Features are cross validated between the two halves of the training data. The next sections will explain how each of the steps in the flowchart was approached, and how problems were solved.

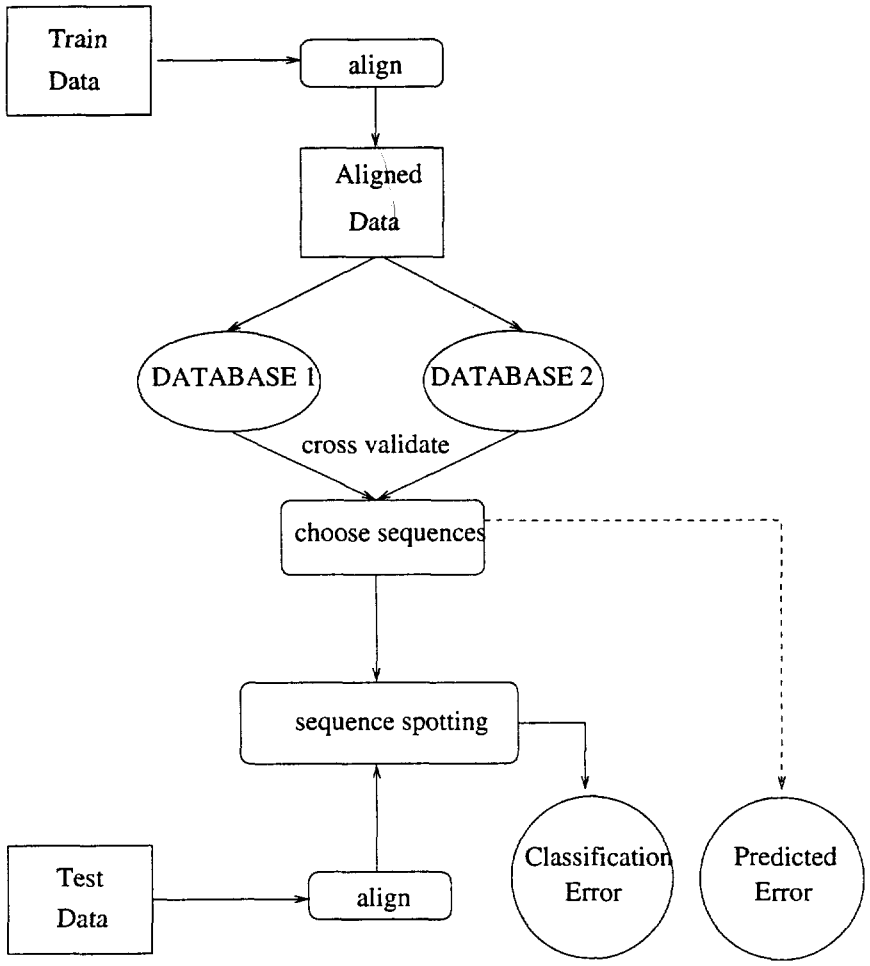

Figure 5.3: Flowchart of the Language Identification System.

5.4.2 Sequence Selection

Selecting sequences by using our theoretical error estimate allows us to add any number of features to the final list until the discrimination of the training set is estimated to be zero. However, there is no mechanism to prevent overtraining. Two methods to avoid overtraining are adopted: choosing a small set of features and cross validation which will be explained in Section 5.5.2. Figure 5.4 shows several curves indicating the theoretical estimate of the error as a function of the number of segments processed. Each performance curve corresponds to a given number of sequences as features. It can be seen that the

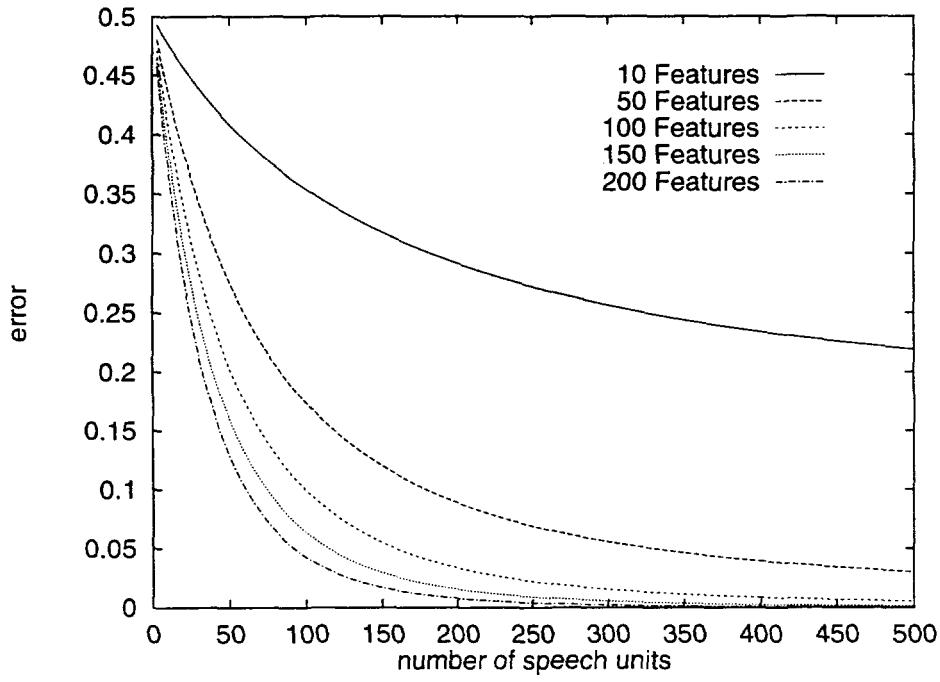

Figure 5.4: Estimated error as a function of time and number of sequences used as features

increase in performance when going from 10 to 100 features is much greater than the increase when going from 100 features to 200 features, a good indication that overtraining might occur as the number of features increases much above 100. Experimental results in Section 5.5.2 confirm this. In addition, we compared sets of features derived from files aligned with 59 phoneme classes against those aligned with 95 phonemes. After clustering we are able to select significantly more sequences with lower estimated error as shown in Figure 5.5.

5.4.3 Sequence Spotting

Sequence spotting is performed by representing all sequences in form of a tree. Thus, for each new incoming sequence from the incoming utterance a pointer into the tree is started. At the same time all existing pointers are either advanced in the tree if the branch matches the incoming segment label, or terminated if there is no match. Each branch of the tree is marked with a label signifying whether or not it constitutes the ending of a sequence within the tree. For all pointers which reach such an end, the frequency count

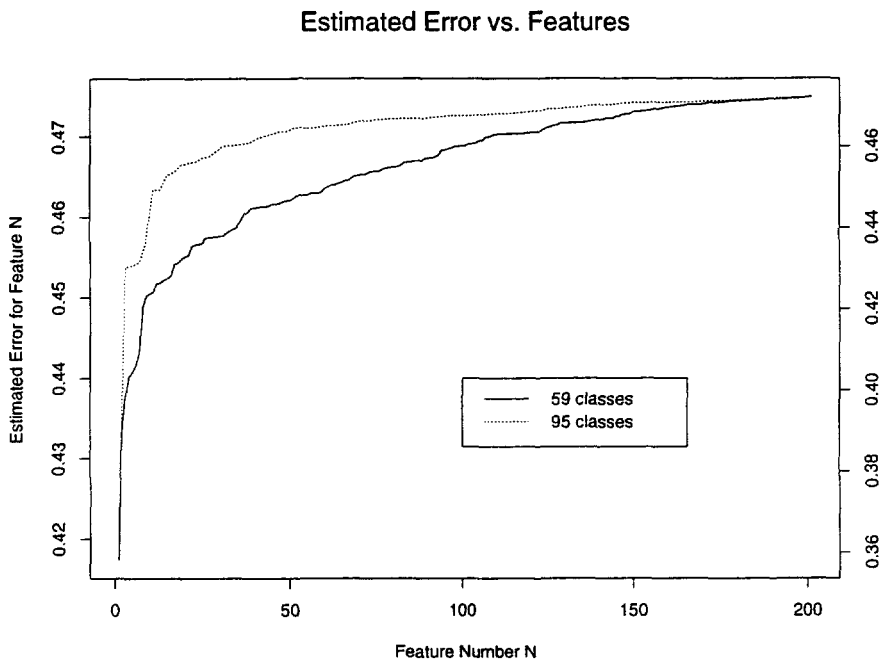

Figure 5.5: Plot of estimated error for each sequence corresponding to the sorted list.

of the corresponding sequence is incremented. More than one sequence can end at a given time and more than one sequence can start at a given time. The advantage of a tree representation is clearly the time saved in evaluating sequences which have common prefixes.

5.4.4 Language Identification

We compare two methods of language classification. The first uses parametric statistics, while the second uses neural networks. Assuming normal distributions the first method is used as it was in Section 3.4 for estimating language identification error at each stage of merging. We use this method primarily to indicate the close relation between theoretical

estimation and true language identification. The second method using neural networks allows us to combine all features in a non-linear fashion, appropriately handling the co-occurrence of features and making no assumption about the feature distribution.

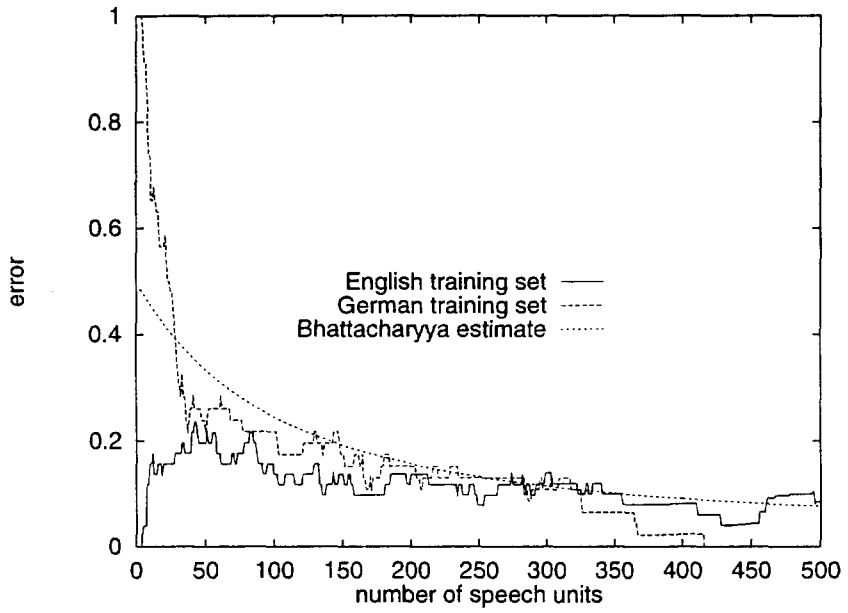

Figure 5.6: Estimate and Actual Classification Error Probability as a Function of Number of Processed Speech Segments Using Statistical Approach.

Statistics-based Language Classification Language classification is based on the occurrence frequencies of selected sequences in the output of the first module. The statistical approach is consistent with the feature selection process. By assuming normal distributions, we let y equal the weighted scalar product of the feature vector $\vec{u}$ and the weight vector $\vec{\alpha}$

$$y = \sum_j \alpha_j u_j \tag{5.3}$$

where u_j is the frequency of sequence j in a given aligned file. Then this file is classified as:

$$\underset{\forall i \in lang}{argmax}\{\frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}[\frac{y-\mu_i}{\sigma_i}]^2}\} \tag{5.4}$$

Figure 5.6 plots the fraction of correctly classified sequences for the English and German training set as a function of time. It is important to note that the theoretical prediction represents a valid estimate of the actual language classification error on the training set, as shown here. However, as the number of features increases this estimate is no longer valid in the same degree that co-occurrence increases. The theoretical model does not take this into account. This method is therefore constrained to using a small number of features.

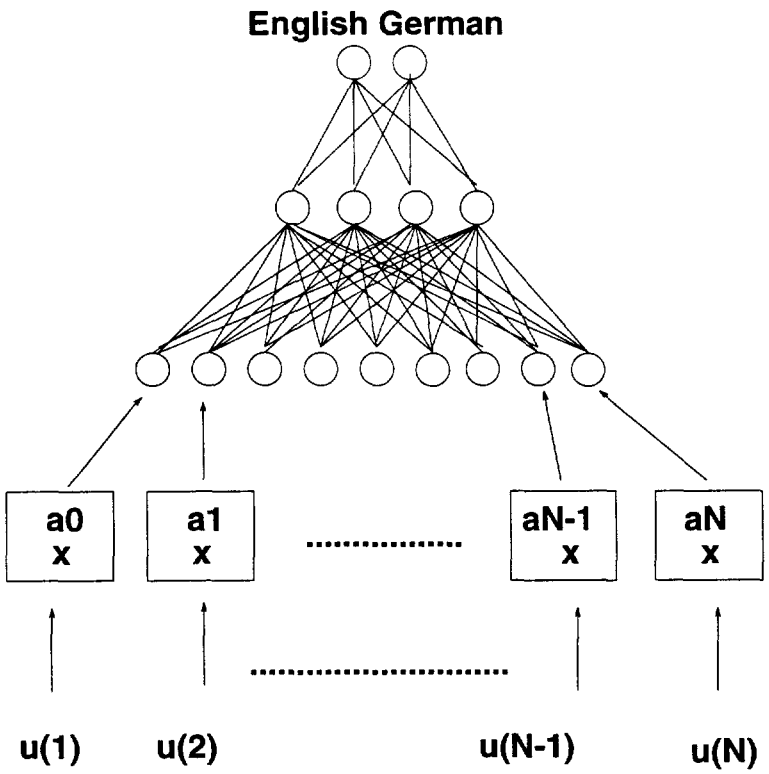

Figure 5.7: Neural network based setup for language identification

Non-linear Language Classification Since the normal assumption may not be appropriate, we have also used a non-linear neural network as classifier. Such a classifier is able to take correlation between features into account. For each of the sequences to be

spotted, the occurrence frequency of each selected sequence, normalized by the number of labels seen at the time of classification (possibly multiplied by the appropriate α) is used as a feature. This set is used to train a neural network in order to learn the discriminant function.

5.5 Results

In this section we wish to show that clustering of phonemes has advantages.

1. We obtain better alignment with the merged phoneme classes.

2. As a result, language identification improves.

By decreasing the number of phonemes to be recognized the phoneme recognition accuracy and alignment were improved as shown in Section 5.2. Language identification is directly affected by this improvement.

5.5.1 Statistics-based Language Classification

Statistics-based language identification is performed with a small number of 30 features in order to suppress a high degree of co-occurrence. We expect deviation from our theoretical predictions to the extent that there are correlations between the features and that their distributions are not normal, since these assumptions were used for the calculation of the theoretical error. Appendix C.4 lists the selected sequences. It can be seen that there is a small degree of overlap between the feature strings. Figure 5.6 demonstrates that the theoretical estimate closely matches the actual error rate. This also validates the use of an upper limit given by the Bhattacharyya distance as opposed to the Bayes error which was shown to match closely in theory. Figure 5.8 plots the ratio of the Bhattacharyya distance with respect to the simplified distance measure (Eq. 3.13) used to estimate the discrimination error. The ratio goes to 1 as more sequences are added to the list of features and the simplified measure becomes increasingly accurate. Even for the first feature this measure is already reasonable (ratio of .88). Final results for discrimination based on the normal assumption are given in Table 5.7.

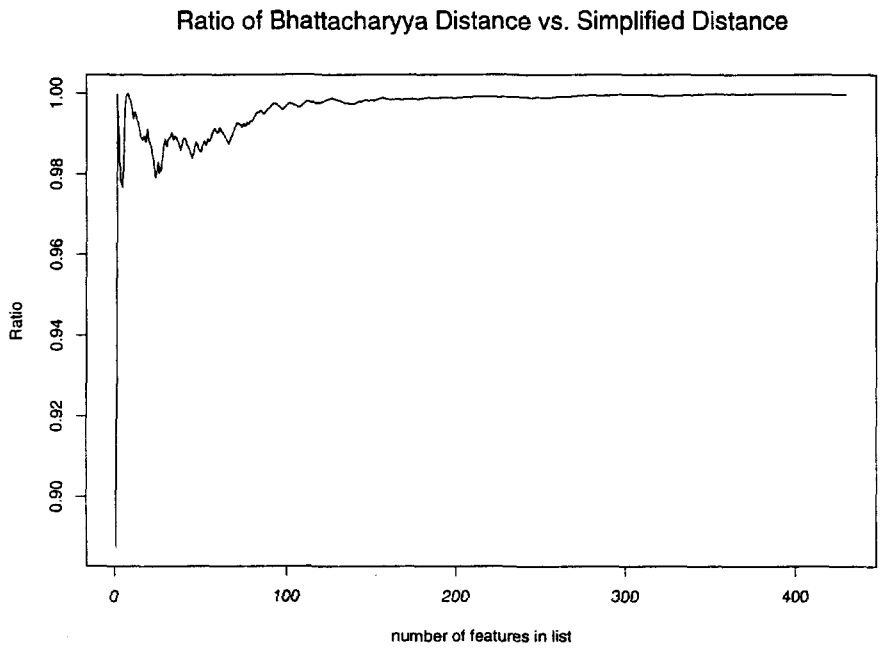

Figure 5.8: Ratio of complete to simplified Bhattacharyya distance measure as a function of the number of sequences in the list.

Table 5.7: Error rates when classification assumes a normal distribution.

Data Set		**Performance after N segments**			
Language	Set	**N=10**	**N=50**	**N=100**	**N=500**
English	Train	0.15	0.20	0.13	0.08
English	Dev	0.11	0.11	0.16	0.10
German	Train	0.73	0.21	0.21	0.00
German	Dev	0.95	0.45	0.30	0.14

5.5.2 Non-linear Language Classification

We train on features derived after having seen 300 phonemes of an utterance to build a representative statistic of the utterance. It was found that this performs better than training on features derived from less mature statistics at earlier points in the utterance. Another practical change to the system regards cross validation during sequence selection. It was found that the method does not generalize well from training to test set. The solution is to split the training set into two parts. In this case ordering of the sequences is performed on half the training set. Error estimates are calculated on both halves and compared.

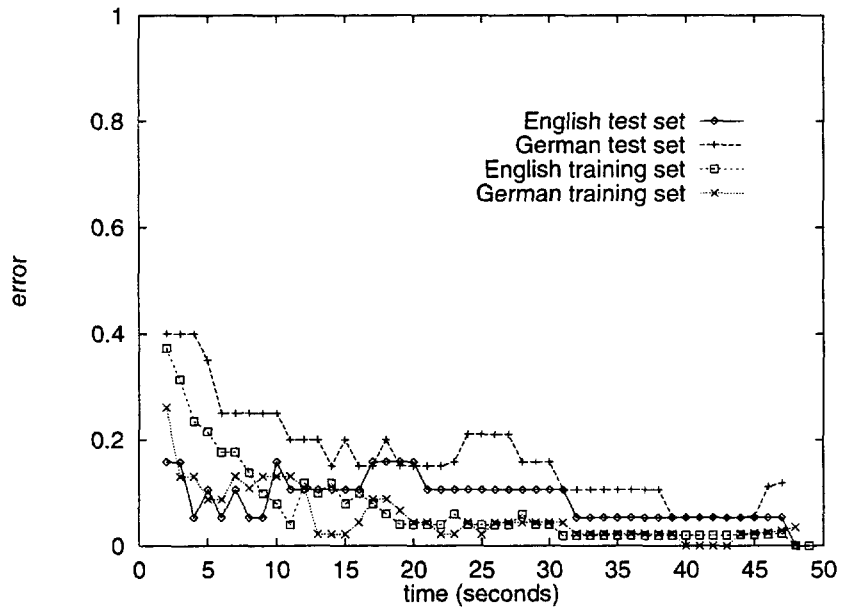

Figure 5.9: Classification Error Probability as a Function of Time Using the Neural Network Approach.

These results show that language identification performance is best when using the neural network approach without weighting sequences and spotting for a set of 100 derived sequences. The first 30 are shown in Appendix C.4. These sequences were chosen out of a list of sequences up to length five. However, no sequences longer then length four occur in the feature set. Figure 5.9 shows the classification rate in identifying English and German as a function of the length (in seconds) of the observed utterance.

Table 5.8: Error rates using neural network with weighting of features

Data Set		Performance after N seconds								
Language	Set	N=5	N=10	N=15	N=20	N=25	N=30	N=35	N=40	N=45
English	Train	0.27	0.08	0.02	0.00	0.00	0.00	0.00	0.00	0.00
English	Test	0.11	0.05	0.16	0.11	0.11	0.11	0.11	0.11	0.05
German	Train	0.17	0.13	0.04	0.02	0.02	0.02	0.00	0.00	0.00
German	Test	0.45	0.30	0.25	0.25	0.26	0.21	0.26	0.15	0.22

Table 5.9: Error rates using neural network without weighting of features ($\alpha_i = 1$).

Data Set		Performance after N seconds								
Language	Set	N=5	N=10	N=15	N=20	N=25	N=30	N=35	N=40	N=45
English	Train	0.26	0.08	0.04	0.04	0.04	0.04	0.02	0.02	0.02
English	Test	0.11	0.16	0.11	0.16	0.11	0.11	0.05	0.05	0.05
German	Train	0.09	0.13	0.02	0.04	0.02	0.04	0.02	0.00	0.02
German	Test	0.35	0.25	0.20	0.15	0.21	0.16	0.11	0.05	0.06

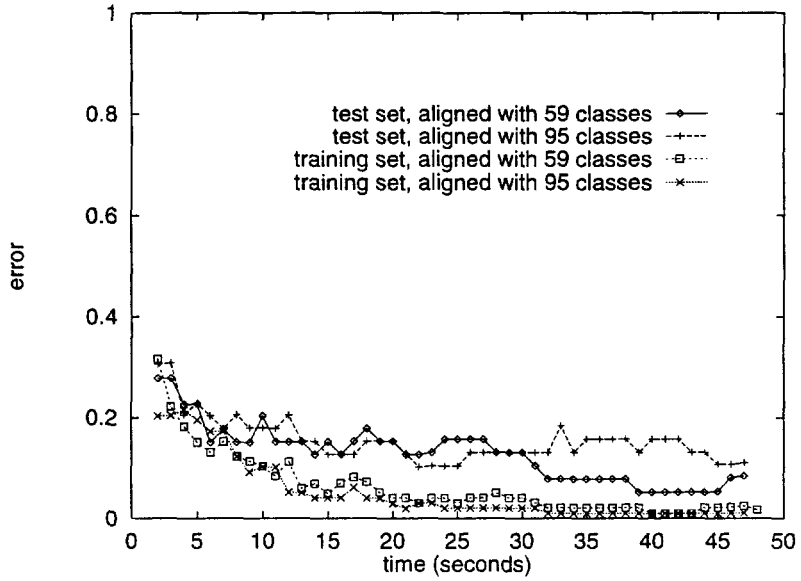

Figure 5.10: Performance before clustering is inferior to performance after clustering.

Results comparing approaches with (Table 5.8) and without using feature weighting (Table 5.9) were obtained. It can be seen that the best results are obtained without weighting the features. This makes sense since the weighting factors are not optimized over the complete set of features. In addition the choice of weights assumes independence of features and normal distributions. The independence assumption is particularly inaccurate, since the sequences and subsequences which make up the feature set are certainly correlated as is evident in Appendix C.4. Figure 5.10 shows the impact of the clustering level on the performance. A small improvement in language classification is evident. Figure 5.11 compares the identification capability of neural networks trained on feature vectors. Each entry in the feature vector represents the normalized occurrence frequency of a discriminating sequence in the file from which the vector is derived. From the training set of the files aligned with 95 phonemes, we have derived 17109 training vectors and 5837 test vectors. From the training set of the files aligned with 59 phonemes, we have derived 20850 training vectors and 7521 test vectors. In Figure 5.11 the percentage of correctly identified feature vectors is plotted as a function of the size of the vector reflecting the number of sequences which are spotted. It is an indication that using the top 100 sequences

selected from the files aligned with 59 phoneme classes gives optimal performance. It can also be seen that clustering of phonemes improves performance consistently. After clustering a higher performance can be obtained for a smaller number of features. The final neural network for discriminating English and German is based on 100 features and has 11 hidden nodes using 59 phoneme classes to align the speech and 15 hidden nodes using 95 phoneme classes to align the speech.

NIST 1994 Evaluations The results obtained here compare favorably with results obtained in the research community on the $OGI - TS$ database as shown in Table 5.5.2. Since 1994 some progress has been made improving on these results. It is however very difficult to compare to newer systems because they are either not tested on a common test set of they are not evaluated on a pairwise basis. Appendix D indicates which files were misrecognized by our system. The probability is 84% that Lincoln Lab's system is significantly better than our system and 98% that our system is significantly better than Lockheed's. Using the two tailed McNeemer test, the probability that our systems using 95 or 59 phonemes make the same mistakes is 38%. They are not significantly different.

Table 5.10: Comparative Results on standard NIST 1994 test set

Test Site	% correct
Lincoln Labs	97
OGI Y. Yan	92
ITT	92
OGI Berkling	92
MIT	84
Lockheed	76

NIST 1996 Evaluations There are two reasons why we would like to report results for a subset of the 1996 NIST evaluations. First, we would like to show that the algorithm is not dependent on our derived set of tokens but is instead generalizable to any set of tokens. Second, these results are compared to those of Yonhong Yan who was the winner of the 1996 evaluations. While the NIST 1994 evaluation contained monologue speech,

the data for the NIST 1996 evaluation contains conversational speech merged over several different databases. These were switchboard, king (wideband and narrowband) and the $OGI - TS$ data. Out of the 22 languages supplied as training and test sets we are only concerned with the two languages, English and German. The training set consisted of 896 English files and 877 German files. For the development test set 240 English files and 240 German files were supplied, divided into parts of 80 files, each of which are 3, 10, and 30 seconds long. These files were aligned with the set of 40 English phonemes with 46.8% accuracy. The optimal list of discriminating features, chosen based on the aligned training and development test sets, consisted of 200 sequences up to length 4. The neural network trained on these features had 73 hidden nodes and performed with 83% accuracy. The final test set contained 478 English and 80 German aligned files which are 30 seconds long. Table 5.5.2 shows the comparative results for our system and Yan's system. Using the two-tailed McNeemer test, it can be shown that the probabilities that the systems make the same errors classifying German and English are 61% and 39% respectively. Figure 5.12 shows the error curve for identifying German and English as a function of time. Even though our system is comparable to Yan's system, it should be pointed out that his system is based on a 22 language evaluation and was not specifically trained to discriminate the two languages English and German.

Table 5.11: Comparative Results on standard NIST 1996 test set

Lang.	**Error**	**System**
EN	5%	(Yan)
GE	11%	(Yan)
EN	3%	(Berkling)
GE	15%	(Berkling)

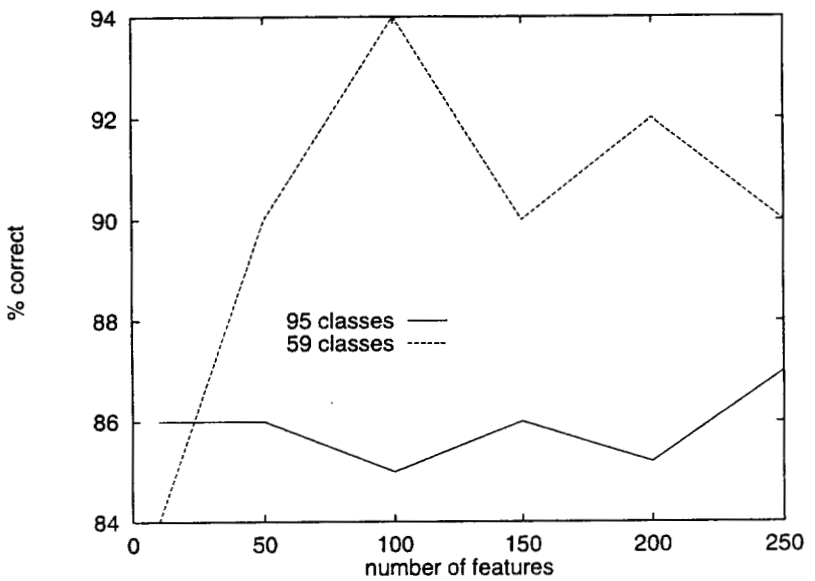

Figure 5.11: % Correct classification for neural netword classifier as a function of the number of features used. Results shown before (95 classes) and after (59 classes) clustering for test set of feature vectors.

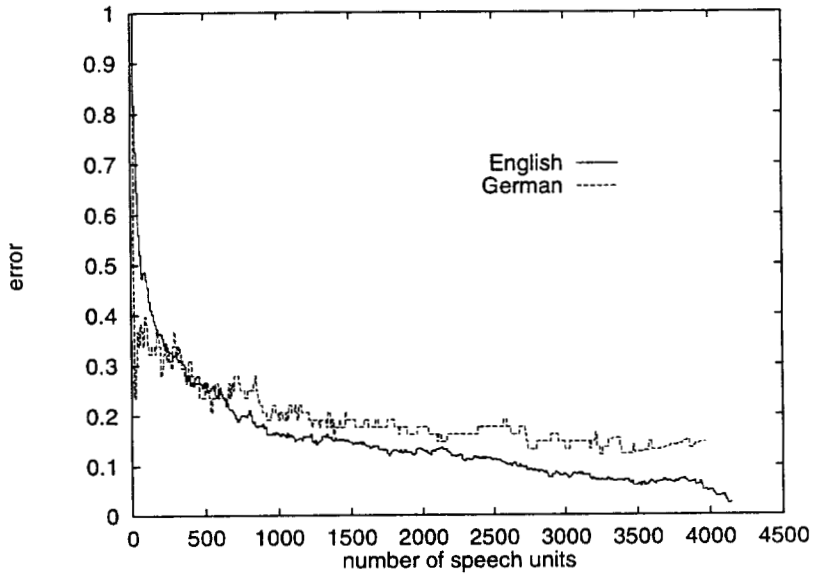

Figure 5.12: The error curve for identifying German and English as a funcition of the time (in ms), where each phoneme segment is artificially set to 10ms.

5.5.3 Evaluating the Impact of Alignment

We have shown that clustering phonemes across languages improves language identification in our system. We achieve this by improving our alignment and thus allowing a larger number of features to be derived. However, the results leave much room for improvement. In this section we want to study the impact that the bad alignment has on our language identification performance.

In order to test the impact of the alignment accuracy on the algorithm presented we artificially create errorful "aligned" files by using the confusion matrix and the labeled files. Changing the degree of correct alignment on a segment-by-segment basis we create files that range from perfect alignment to an alignment with the same accuracy as obtained by the system. We test five degrees of alignment which are derived as follows. The original confusion matrix comparing substitutions between aligned and labeled files will be mutated. This is done by increasing the diagonal entries of the matrix by a given percentage while decreasing the off-diagonal values by the same percentage, thereby increasing the correct alignment. The labeled files are then recast into new labels with the probability given in the new matrix. The formal description of the process follows: let $p(L)$ denote the probability of label L occurring in the labeled files. Let Q denote the matrix of confusion between the labeled and aligned files. Then $q(L)$ is the probability of L occurring in the aligned files given Q and p.

Table 5.12: Simulating Better Alignment.

$q(Newlabel)$	$= Q(Newlabel \mid Oldlabel) * p(Oldlabel)$	
where,		
$Q(Newlabel \mid Oldlabel)$	$= P(Newlabel \mid Oldlabel) + (\sum_{x \neq Oldlabel} (N) P(x \mid Oldlabel))$	; $Newlabel = Oldlabel$
$Q(Newlabel \mid Oldlabel)$	$= (1 - N) P(Newlabel \mid Oldlabel\ )$	; $Newlabel \neq Oldlabel$

We let N in Table 5.12 vary from 1.0 to 0.0 to increase the correct alignment. When $N = 1.0$ the original confusion matrix is used to create data which simulates the alignment which we currently have obtained with our system. The frame-based accuracy is 25% as given in Table 5.3 for this data, which was aligned with 59 phoneme units using a German-English grammar. When $N = 0.0$, the confusion matrix is the identity matrix and the simulated alignment is equal to the labeled files. Data is generated from the labeled files for each value of N and the same process of data-driven error estimation as followed in Section 5.4 is performed in order to automatically derive a list of sequences on which the language discrimination estimate is based. For the top 10 words, Table 5.13 shows different values of accuracy after alignment corresponding to the various levels for N.

Table 5.13: Average Accuracy of phoneme classes corresponding to different N. $N = 1.0$ corresponds to the original aligned data with 25% accuracy after alignment.

N	Average Accuracy(%)
0	100
.2	85
.6	55
.8	40
1.0	25

After generating statistics on occurring sequences, a set of features is selected. We then perform language identification on an equally mutated test set of files. It was already shown that a theoretical estimate closely matches the actual performance on real data. In order to indicate how much improvement in the alignment is necessary for a desired performance we need only to plot the error curve calculated for the selected features at each level of N. For $N = .8$ Table 5.13 shows that the frame based accuracy of the alignment should be 40%. The corresponding curve in Figure 5.13 shows that such a level of accuracy in the alignment would enable us to obtain a significant improvement in language discrimination. Of course, this is a simplified model of misclassification. In practice, errors are much more correlated than in our model. These results are nevertheless suggestive of what can be achieved with better alignment of the data.

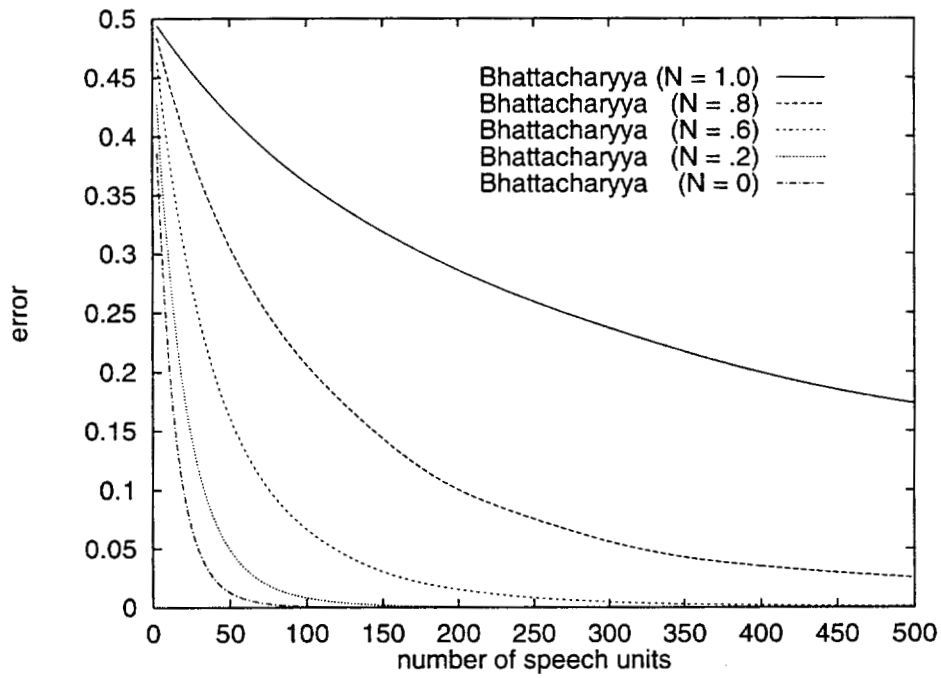

Figure 5.13: Effect of alignment on language identification error: for different alignment accuracies, the classification rate is shown as a function of the number of phonemes in the classified utterance.

Chapter 6

Conclusion

The original goal of the research was to demonstrate that one can build a language identification system that is linguistically sound [1]. At the same time our goal was to study the limits of language discriminability and combine theory with implementation to build a performance-oriented system of minimal complexity. Section 6.1 describes how we accomplished these goals. In Section 6.2, we would like to show why we believe these contributions will be an important part of future systems.

6.1 Summary of Thesis

Although much progress has been made in the past few years, language-identification systems generally tend to be engineered rather than scientifically designed. As a result such systems have not necessarily been linguistically motivated. In this thesis the subject was approached with the belief that features used to discriminate between languages should be linguistically valid. Our premise was that a practical implementation of such a design hinged on the development of a mathematical model to approximate language discrimination. In our final system, we uniquely combined linguistic design, theoretical development and practical system implementation.

It is a well-known fact that humans can identify their own language more quickly and reliably than can any automatic system available today. This fact lead us to attempt a partial reconstruction of the complex human process of language identification. Even a

[1] As an example, we are not using Japanese phonemes in order to discriminate between English and Spanish.

"partial reconstruction" confronts us with pragmatic issues such as the consistent tradeoff between complexity and performance. The complexity of each component in the system should therefore be minimized. Minimization is examined at two levels. First, at the level of speech representation (Section 3.1) and second, at the level of feature extraction (Section 3.2 - 3.3). This results in the implementation of a "word"-spotting algorithm (Chapter 5).

It has been shown in the past that good phoneme recognition directly affects the success of language identification [102]. In order to obtain good phoneme recognition, a meaningful phoneme representation is often sacrificed; for example choosing to decode speech with phoneme set from language A to discriminate between languages B and C because this results in the best performance. The goal in this thesis was to find a set of phoneme-like tokens to represent speech in a linguistically meaningful way while preserving performance (see also [49]). A token set of common phoneme-like speech units across languages was created by taking the union of phonemes from all of the languages that the system is trained on. The resulting complexity was systematically reduced. Since it is impractical to implement a language-identification system for each possible combination of tokens, this thesis introduced a theoretical estimate of language discrimination based on the different choices of token sets (Section 3.5). As a result, it is now possible to design a set of tokens that are detailed enough to capture language-specific phonemes and, at the same time, that are general enough to represent all language sounds in the system (Section 3.4).

Compensating for the inaccuracies of token recognition and alignment is a significant research problem. In order to account for the variability of "Words" in the automatic token alignment, the degree to which the variability within language is less than the variability across languages is determined by statistical analysis (Sections E.1- E.5). A mathematical model was developed in this thesis to understand the impact of inaccurate automatic alignment on language discrimination (Chapter 4). This theory helped explain why modeling within-language variability may not improve the overall discrimination results (Section 4.4). Our findings have held true for the implementation presented in this thesis (Section E.7). The final set of discriminating "words", each standing for a single sequence, seem to represent in part the grammatical inflections in the languages and their

phoneme inventory (Appendix C.4). Based on a perceptual study, these results seem to agree to a non-trivial extent with the way humans identify unknown languages. The final system design is depicted in Figure 6.1. The chosen modules, indicated by arrows, are flexible to capture the inherent and discriminating features of different languages.

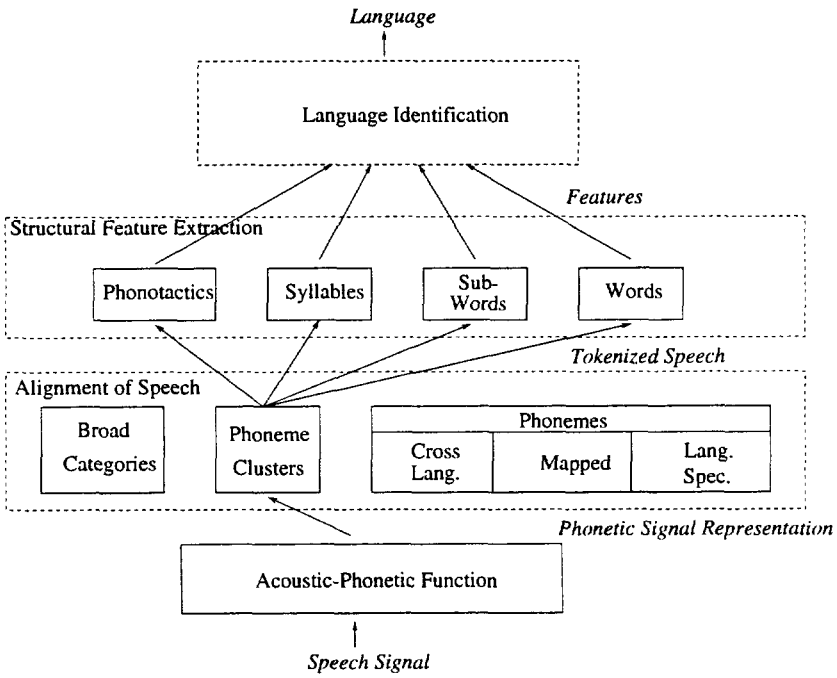

Figure 6.1: Modules of the LID System chosen for this thesis

6.2 Present and Future of Language Identification

In this section we would like to show how system design has evolved in recent years and how we believe that the algorithms developed here conform to some requirements of the next-generation of language identification systems.

6.2.1 Design Issues for Language Identification Systems

One can imagine the complexity of an automatic language identification (ALI) system, outlined in Section 1.1.3, that incorporates language-dependent structural properties of

speech including phonotactics, syllable, sub-words and words. To minimize complexity, any system will be evaluated with respect to basic design criteria, including those given in Table 6.1, thus, such a "perfect" system may actually be poorly suited for some applications.

Table 6.1: General Evaluation Criteria of Language Identification Systems

1) Performance
2) System complexity
3) Training requiring labeled data
4) Extensibility

Performance is one of the most important criteria for the evaluation of implemented systems, as evaluated in terms of the accuracy and speed of the ALI. Performance is partly a function of the length of utterance needed to identify the language of the utterance. System complexity is also important, even if it may play a decreasing role in the future due to improved computing speed and larger number of available computers. Finally, there is a great need to minimize the requirements of labeled data to train a system. Even though the availability of larger databases has increased, data are not usually labeled at the phonemic level or even at the word level. The system must compensate for this. Recent advances in LID have addressed issues such as automatically labeling data with phoneme-like units [66]. Databases have grown from including four languages to ten, twenty, and more. Systems will most likely generalize to a large number of languages in the future.

Yearly system evaluations are hosted by the National Institute for Standards in Technology (NIST), where labs from all over the world are evaluated on a common task. From year to year this task has increased in complexity. The extended list of criteria for 1996 is shown in Table 6.2. The first criterion refers to task independence. While being an important criterion, performance figures may be deceptive and task dependent. Even though

a system may have been engineered for a particular task, this same design may not be generalizable despite a good identification rate for a given task. The second criterion refers to the discrimination of similar languages. Because the number of languages to be identified will grow, their similarity tends to increase to the degree that they can even be considered dialects of one another. The new design criteria reflect applications which discriminate between a large number of languages and dialects by stating that systems should be robust enough to achieve uniform performance across all the languages to be identified. In the future, systems will be used in a variety of different tasks, including applications to improve robustness of speech recognition systems for dialect and accent detection. Preliminary systems are already in place [110].

Table 6.2: Evaluation Criteria of Language Identification Systems (NIST 1996)

1) Task Independence
2) Discrimination between Similar Languages
3) Discrimination between Dialects
4) Achieving Uniform Performance Across Languages

6.2.2 This Thesis and the Future

Some of the material presented in this dissertation is attractive according to these NIST design criteria for 1996. Specifically:

1. **Task Independence**

   The algorithms developed in this thesis are not dependent on any particular application. There are only two restrictions imposed on their application. First, the algorithm is only implemented for a two-class problem. However, one can easily extend the algorithm since all the statistical measures generalize to multiple classes.

Second, the classes to be discriminated are represented by streams of tokens. This is not a severe restriction in that most systems today represent speech in terms of some sort of token (such as phonemes) before analyzing it. There is no restriction on the type of token used so long as the set of tokens are common across the classes to be discriminated.

2. **Discrimination between Similar Languages**

   The algorithm presented in this thesis automatically selects features based on their frequency of occurrence and their distributions in the two classes to be discriminated. No restriction is imposed on the length of the sequence of tokens to be selected as a feature. This is why the system generalizes well to discriminate between similar languages. Some of the problems with similar languages may be the small number of features and the particular type of features that discriminate them. The large variance in language dependent structural features makes it difficult to choose the correct system design (ranging from phoneme-spotting to large vocabulary systems) by hand. The algorithm introduced in this thesis automatically selects features without restriction to solve the particular problems of any given language pair.

3. **Discrimination between Dialects**

   Dialects refer to the words, language forms, pronunciations, and speech habits peculiar to people of specific geographic regions [14]. The reason our system is flexible enough for this type of application is similar to item 2 above because there is a fine line between similar languages and dialects. In addition, we provided the framework for automatically designing a token set to highlight the pronunciation differences of specific phonemes.

4. **Achieving Uniform Performance Across Languages**

   Language identification systems in place today rely on a wide variety of features, ranging from phoneme occurrence, to phonotactics, to large vocabulary. Such systems require a design choice for the type of feature to be made independently of the languages in the system. However, each language has inherent particularities such as the consonant-vowel structure of Japanese, the short but frequent grammatical

inflections in English, or the guttural fricatives in German (see Section 1.2.1). Making design decisions independent of the languages in the system may not allow the necessary flexibility needed to capture all the language specific features. In other words, languages vary to different degrees and a design that works for one language may not work for another. In this thesis, discriminating "words" of any length are selected from a representation of speech using customized tokens. This process is automatic and allows a flexible design appropriate for the specified languages in the system.

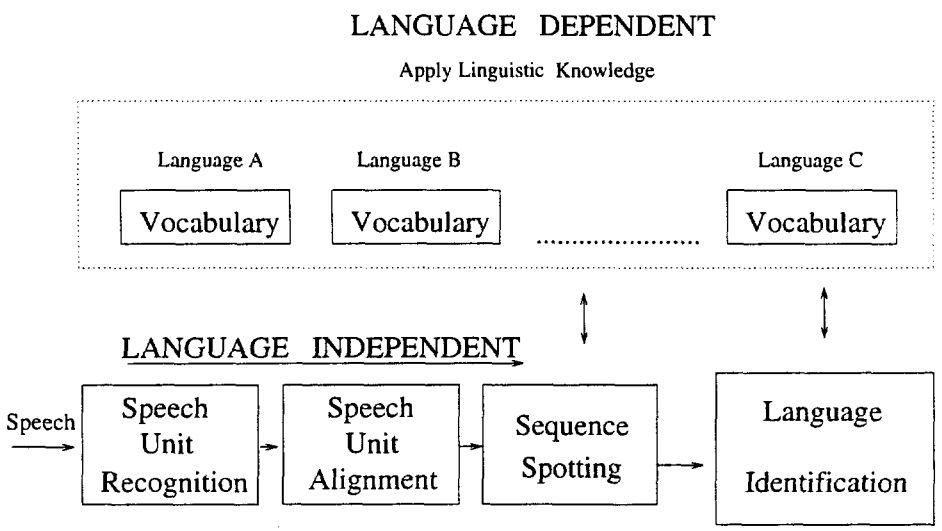

Figure 6.2: Modules of extended LID System The system consists of a phoneme recognizer, followed by an automatic alignment of the speech with the recognized phonemes. Structural features for each language are derived based on their discriminating sequences.

The system introduced in this thesis can be generalized to any number of languages as shown in Figure 6.2. By discriminating a given language with respect to all other languages in the system, a vocabulary capturing both its inherent and discriminating structure can automatically be derived. Similarly, the application can be used in conjunction with the parallel approach to language identification. This is the most successful type of system implemented to date. In such a system, the utterance is decoded in parallel by several language dependent phoneme recognizers. When applying our algorithm to this setup, the

only change in the block diagram of Figure 6.2 is the use of language-dependent speech unit alignment.

Although the resulting system does not currently perform as well as the very best systems for language identification [102, 112], we believe that it is interesting in its own right. As the accuracy of speech recognition improves, it may also be that a system such as ours improves more substantially than those based on, say, phonotactic statistics. It has recently been pointed out that the path to improved speech-recognition systems does not necessarily imply improved performance at every step along the way [12, 13]. Instead, some steps need to be taken even if they increase error rates; other criteria (such as robustness, elegance, or extendability) should be used to evaluate such steps. We believe the same to be true in language identification. Although the sytem proposed in this thesis does not improve on state-of-the-art performance, it introduces several novel capabilities which may be of importance in the long run.

Bibliography

[1] ANDERSEN, O., DALSGAARD, P., AND BARRY, W. On the use of data-driven clustering technique for identification of poly- and mono-phonemes for four european languages. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Adelaide, Australia, April 1994), vol. 1, pp. 121 - 124.

[2] ANDERSEN, O., DALSGAARD, P., AND BARRY, W. Data-driven identification of poly- and mono-phonemes for four european languages. In *Proceedings Eurospeech 93* (Berlin, Germany, September 1993), vol. 2, pp. 759–762.

[3] ANDERSEN, O., DALSGAARD, P., AND HANSEN, A. Multi-lingual testing of a self-learning approach to phonemic transcription of orthography. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 1117–1120.

[4] BARNARD, E., AND COLE, R. A. A neural-net training program based on conjugate-gradient optimization. Tech. Rep. CSE 89-014, Oregon Graduate Institute, 1989.

[5] BARNETT, J., BAMBERG, P., DEMEDTS, A., EVEN, S. V., AND MANGANARO, L. Comparative performance in large-vocabulary isolated word recognition in five european languages. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 1, pp. 189–192.

[6] BARRY, W., AND DALSGAARD, P. Speech database annotation. the importance of a multi-lingual approach. In *Proceedings Eurospeech 93* (Berlin, Germany, September 1993), vol. 1, pp. 759–762.

[7] BEATTIE, V., AND ROHLICEK, J. R. An integrated multi-dialect speech recognition system with optional speaker adaptation. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 1, pp. 1123–1126.

[8] BERKLING, K. M., ARAI, T., BARNARD, E., AND R.A.COLE. Analysis of phoneme-based features for language identification. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Adelaide, Australia, April 1994), vol. 1, pp. 289 - 292.

[9] BERKLING, K. M., AND BARNARD, E. Theoretical error prediction for a language identification system using optimal phoneme clustering. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 1, pp. 351–354.

[10] BERKLING, K. M., AND BARNARD, E. Language identification with inexact sequence matching. In *Proceedings International Conference on Spoken Language Processing* (Philadelphia, USA, October 1996), pp. 1796–1799.

[11] BERKLING, K. M., AND BARNARD, E. Language identification with multilingual phoneme clusters. In *Proceedings International Conference on Spoken Language Processing* (Yokohama, Japan, September 1994), vol. 4, pp. 1891–1894.

[12] BOURLARD, H. Towards increasing speech recognition error rates. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 883–894.

[13] BOURLARD, H., HERMANSKY, H., AND MORGAN, N. Towards increasing speech recognition error rates. *Speech Communications 18*, 3 (May 1996), 205–231.

[14] CALVERT, D. R. *Descriptive Phonetics*, 2nd ed. Georg Thieme Verlag Stuttgart, 1986.

[15] CAREY, M., AND PARRIS, E. Topic spotting with task independent models. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 3, pp. 2133–2136.

[16] CAREY, M. J., AND PARRIS, E. S. Topic spotting with task independent models. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 3, pp. 2133–2136.

[17] COLE, R. A., INOUYE, J. W. T., MUTHUSAMY, Y. K., AND GOPALAKRISHNAN, M. Language identification with neural networks: a feasibility study. In *Proceedings IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing* (June 1989), pp. 525–529.

[18] COMRIE, B. *The World's Major Languages*, 1st ed. Oxford University Press, 1990.

[19] COOK, G., AND ROBINSON, T. Utterance clustering for large vocabulary continuous speech recognition. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 1, pp. 141–144.

[20] COVER, T. M., AND THOMAS, J. A. *Elements of Information Theory*, 2nd ed. John Wiley, 1991.

[21] CRYSTAL, D. *The Cambridge Encyclopedia of Language*. Cambridge University Press, 1987, pp. 160–169,280–339.

[22] CUTLER, A., DAVIS, S. M., AND OTAKE, T. Listeners' representations of within-word structure: a cross-linguistic and cross-dialectral investigation. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 3, pp. 1703–1706.

[23] DALSGAARD, P. Phoneme label alignment using acoustic-phonetic features and gaussian probability density functions. *Computer Speech and Language 6* (1992), 303–329.

[24] DALSGAARD, P., AND ANDERSEN, O. Identification of mono- and poly-phonemes using acoustic-phonetic features derived by a self-organizing neural network. In *Proceedings International Conference on Spoken Language Processing* (Banff, October 1992), vol. 1, pp. 547–550.

[25] DALSGAARD, P., AND ANDERSEN, O. Application of inter-language phoneme similarities for language-identification. In *Proceedings International Conference on Spoken Language Processing* (Yokohama, Japan, September 1994), vol. 4, pp. 1903–1906.

[26] DALSGAARD, P., AND ET AL. On the use of acoustic-phonetic features in ineractive labelling of multi-lingual speech corpora. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (San Francisco, USA, March 92), vol. 1, pp. 549 – 552.

[27] EADY, S. J. Differences in the F_0 patterns of speech: Tone language versus stress language. *Language and Speech 25*, 1 (1982), 29–42.

[28] FANTY, M., POCHMARA, J., AND COLE, R. An interactive environment for speech recognition research. In *Proceedings International Conference on Spoken Language Processing* (Banff,Alberta,Canada, October 1992), vol. 2, pp. 1543–1547.

[29] FOIL, J. T. Language identification using noisy speech. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (April 1986), vol. 2, pp. 861–864.

[30] FUKUNAGA, K. *Introduction to Statistical Pattern Recognition*, 2nd ed. Academic Press, Inc., 1990.

[31] GISH, H., NG, K., AND ROHLICEK, R. Secondary processing using speech segments for an hmm word spotting system. In *Proceedings International Conference on Spoken Language Processing* (Banff,Alberta,Canada, October 1992), vol. 1, pp. 17–20.

[32] GOESCHEL, J. Artikulation und Distribution der sogenannten Liquida r in den Europaeischen Sprachen. *Indogermanische Forschungen 76* (1971), 84–126.

[33] GOODMAN, F. J., MARTIN, A. F., AND WOHLFORD, R. E. Improved automatic language identification in noisy speech. In *International Conference of the American Society for Signal Processing* (1989), pp. 528–531.

[34] GROVER, C., JAMIESON, D. G., AND DOBROVOLSKY, M. B. Intonation in English, French and German: perception and production. *Language and Speech 30*, 3 (1987), 277–295.

[35] HANLEY, T. D., SNIDECOR, J. C., AND RINGEL, R. L. Some acoustic differences among languages. *Phonetica 14*, 2 (1966), 97–107.

[36] HAZEN, T. J. *Automatic Language Identification Using a Segment-based Approach.* Master's thesis, Massachusetts Institute of Technology, Aug. 1993.

[37] HAZEN, T. J., AND ZUE, V. W. Recent improvements in an approach to segment-based automatic language identification. In *Proceedings International Conference on Spoken Language Processing* (Yokohama, Japan, September 1994), vol. 4, pp. 1883–1886.

[38] HERMANSKY, H. Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustic Society of America 4* (April 1990), 1738–1752.

[39] HIERONYMUS, J. L. ASCII phonetic symbols for the world's languages: Worldbet. Tech. rep., AT&T Bell Laboratories, Murray Hill, NJ 07974 USA, 1994. ftp://speech.cse.ogi.edu/pub/docs/worldbet.ps.

[40] HOUSE, A. S., AND NEUBERG, E. P. Toward automatic identification of the language of an utterance: Priliminary methodological considerations. *Journal of the Acoustical Society of America 62*, 3 (1977), 708–713.

[41] IRII, H., ITOH, K., AND KITAWAKI, N. Multilingual speech database for evaluating quality of digitized speech. In *Proceedings International Conference on Spoken Language Processing* (Kobe, Japan, 1990), vol. 2, pp. 1025–1028.

[42] ITAHASHI, S., AND DU, L. Language identification based on speech fundamental frequency. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 1359–1362.

[43] ITAHASHI, S., ZHOU, J., AND TANAKA, K. Spoken language discrimination using speech fundamental frequency. In *Proceedings International Conference on Spoken Language Processing* (Yokohama, Japan, September 1994), vol. 4, pp. 1899–1902.

[44] JAKOBSON, R., FANT, C. M., AND HALLE, M. *Preliminaries to Speech Analysis. The Distinctive Features and their Correlates*, 7th ed. The MIT Press, 1967.

[45] JELINEK, F. *Readings in Speech Recognition.* Morgan Kaufman Publishers Inc., San Mateo, California, USA, 1990, Ch. Language Processing for Speech Recognition, pp. 450–507.

[46] JONGENBURGER, W., AND HEUVEN, V. V. The role of linguistic stress in the time course of word recognition in stress-acent languages. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 3, pp. 1695–1698.

[47] KADAMBE, S., AND HIERONYMUS, J. L. Spontaneous speech language identification with a knowledge of linguistics. In *Proceedings International Conference on Spoken Language Processing* (Yokohama, Japan, September 1994), vol. 4, pp. 1879–1882.

[48] KOTANI, M., AND MATSUMOTO, H. Sound perception between two languages based on analyses of onomatopoeic expression. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 3, pp. 2263–2266.

[49] KWAN, H., AND HIROSE, K. Recognized phoneme-based n-gram modeling in automatic language identification. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 1367–1370.

[50] LADEFOGED, P. Discussion note. the revised international phonetic alphabet. *Language 66*, 3, 550–552.

[51] LADEFOGED, P. The revised international phonetic alphabet. *Journal of the International Phonetic Association 19*, 2 (1990), 67–80.

[52] LADEFOGED, P. *A Course in Phonetics*, 3rd ed. Harcourt Brace Jovanovich, 1993.

[53] LAMEL, L., AND GAUVAIN, J.-L. A phone-based approach to non-linguistic speech feature identification. *Computer Speech and Language 9*, 1 (January 1995), 87–105.

[54] LAMEL, L., GAUVAIN, J.-L., AND ADDA-DECKER, M. Issues in large vocabulary. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 1, pp. 185–189.

[55] LAMEL, L. F., AND GAUVAIN, J.-L. S. Language identification using phone-based acoustic likelihoods. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Adelaide, Australia, April 1994), vol. 1, pp. 293 – 296.

[56] LAMEL, L. F., AND GAUVAIN, J.-L. S. Identifying non-linguistic speech features. In *Proceedings Eurospeech 93* (Berlin, Germany, September 1993), vol. 1, pp. 23–30.

[57] LAMES, L. F., AND GAUVAIN, J. L. Language identification using phone-based acoustic likelihoods. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Adelaide, Australia, April 1994), vol. 1, pp. 293 - 296.

[58] LANDER, T., AND METZLER, T. The cslu labeling guide. Tech. Rep. CSLU 003, Oregon Graduate Institute, 1994.

[59] LEONARD, R. G. Language recognition test and evaluation. Tech. Rep. RADC-TR-80-83, Air Force Rome Air Development Center, March 1980.

[60] LEONARD, R. G., AND DODDINGTON, G. R. Automatic language identification. Tech. Rep. RADC-TR-74-200, Air Force Rome Air Development Center, August 1974.

[61] LEONARD, R. G., AND DODDINGTON, G. R. Automatic language discrimination. Tech. Rep. RADC-TR-78-5, Air Force Rome Air Development Center, January 1978.

[62] LEONARD, R. G., AND DODDINGTON, G. R. Automatic language identification. Tech. Rep. RADC-TR-75-264, Air Force Rome Air Development Center, October 1975.

[63] LI, K., AND EDWARDS, T. Statistical models for automatic language identification. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (April 1980), pp. 884–887.

[64] LI, K. P. Automatic language identification using syllabic spectral features. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Adelaide, Australia, April 1994), vol. 1, pp. 297 - 300.

[65] LUND, M., AND GISH, H. Two novel lanuage model estimation techniques for statistical language identification. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 1363–1366.

[66] LUND, M., MA, K., AND GISH, H. Statistical language identification based on untranscribed training. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Atlanta, USA, May 1996), vol. 1, pp. 793–796.

[67] MARTIN, S., LIERMANN, J., AND NEY, H. Algorithms for bigram and trigram word clustering. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 1253–1256.

[68] MARZAL, A., AND VIDAL, E. Computation of normalized edit distance and applications. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (September 1993), vol. 15, pp. 926–932.

[69] MATSUNAGA, S., SINGER, H., AND MATSUMURA, T. Continuous speech recognition using non-uniform unit based acoustic and language models. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 3, pp. 1619–1622.

[70] MCDONOUGH, J., AND GISH, H. Issues in topic identification on the switchboard corpus. In *Proceedings International Conference on Spoken Language Processing* (Yokohama, Japan, October 1994), vol. 4, pp. 2163–2166.

[71] McDonough, J., and Ng, K. Approaches to topic identification on the switchboard corpus. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Adelaide, Australia, April 94), vol. 1, pp. 385–388.

[72] Mendoza, S., Gilick, L., Ito, Y., Lowe, S., and Newman, M. Automatic language identification using large vocabulary coninuous speech recognition. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Atlanta, USA, May 1996), vol. 1, pp. 785–788.

[73] Moisa, L., and Giachin, E. Automatic clustering of words for probabilistic language models. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 1249–1252.

[74] Muthusamy, Y., Cole, R., and Oshika, B. The OGI multi-language telephone speech corpus. In *Proceedings International Conference on Spoken Language Processing* (Banff,Alberta,Canada, October 1992), vol. 2, pp. 895–898.

[75] Muthusamy, Y. K. A review of research in automatic language identification. Tech. Rep. CS/E 92-009, Oregon Graduate Institute, May 1992.

[76] Muthusamy, Y. K. *A Segmental Approach to Automatic Language Identification.* PhD thesis, Oregon Graduate Institute, July 1993.

[77] Muthusamy, Y. K., Barnard, E., and Cole, R. A. Reviewing automatic language identification. *IEEE Signal Processing Magazine 11*, 4 (October 1994), 33–41.

[78] Muthusamy, Y. K., Berkling, K. M., Arai, T., Cole, R. A., and Barnard, E. A comparison of approaches to automatic language identification. In *Proceedings Eurospeech 93* (Berlin, Germany, September 1993), vol. 1, pp. 1307–1310.

[79] Muthusamy, Y. K., and Cole, R. A. A segment-based automatic language identification system. In *Advances in Neural Information Processing Systems*, J.E.Moody, S.J.Hanson, and R.P.Lippmann, Eds., vol. 4. Morgan Kaufmann, 1992, pp. 241–250.

[80] Muthusamy, Y. K., and Cole, R. A. Automatic segmentation and identification of ten languages using telephone speech. In *Proceedings International Conference on Spoken Language Processing 92* (Banff, Alberta, Canada, 1992 October), vol. 2, pp. 1007–1010.

[81] Muthusamy, Y. K., Cole, R. A., and Gopalakrishnan, M. A segment-based approach to automatic language identification. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Toronto, Canada, May 1991), vol. 1, pp. 353–356.

[82] Muthusamy, Y. K., Jain, N., and Cole, R. A. Perceptual benchmarks for automatic language identification. In *International Conference on Speech and Signal Processing* (Adelaide, Australia, April 1994), vol. 1, pp. 333–336.

[83] Nakagawa, S., Ueda, Y., and Seino, T. Speaker-independent, text-independent language identification by HMM. In *Proceedings International Conference on Spoken Language Processing* (Banff, Alberta, Canada, October 1992), vol. 2, pp. 1011–1014.

[84] NOWELL, P., AND MOORE, R. The application of dynamic programming techniques to non-word based topic spotting. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 1355–1358.

[85] PARRIS, E., AND CAREY, M. Language identification using multiple knowledge sources. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Detroit,USA, May 1995), vol. 5, pp. 3519–3523.

[86] PYE, D., YOUNG, S., AND WOODLAND, P. Large vocabulary multilinual speech recognition using htk. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 1, pp. 181–184.

[87] RAMESH, P., AND ROE, E. B. Language identification with embeddeed word models. In *Proceedings International Conference on Spoken Language Processing* (Yokohama, Japan, September 1994), vol. 4, pp. 1887–1890.

[88] REYES, A. A., SEINO, T., AND NAKAGAWA, S. Two language identification methods based on hmms. In *Proceedings International Conference on Spoken Language Processing* (Yokohama, Japan, September 1994), vol. 4, pp. 1895–1898.

[89] ROGINSKI, K. R. *A Neural Network Phonetic Classifier for Telephone Speech.* Master's thesis, Oregon Graduate Institute, 1991.

[90] SALAVEDRA, J., ZELJKOVIC, I., J.WILPON, RAHMIN, M., AND JACOBSEN, C. Multi-lingual connected digits recognition. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 3, pp. 2119–2123.

[91] SAVIC, M., ACOSTA, E., AND GUPTA, S. K. An automatic language identification system. In *International Conference of the American Society of Signal Processing* (Toronto,Canada, 1991), vol. 2, pp. 817–820.

[92] SCHULTZ, T., ROGINA, I., AND WAIBEL, A. Lvcsr-based language identification. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Atlanta, USA, May 1996), vol. 1, pp. 781–784.

[93] SIU, M., GISH, H., AND ROHLICEK, R. Predicting word spotting performance. In *Proceedings International Conference on Spoken Language Understanding* (Yokohama,Japan, October 1994), vol. 4, pp. 2195–2198.

[94] SOLE, M.-J., AND ESTEBAS, E. Connected speech processes: a cross-linguistic study. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 3, pp. 2239–2242.

[95] STROM, V. Detection of accents. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 3, pp. 2039–2041.

[96] SUGIYAMA, M. Automatic language recognition using acoustic features. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (May 1991), vol. 2, pp. 813–816.

[97] SUGIYAMA, M. Automatic language recognition using acoustic features. Tech. Rep. TR-I-0167, ATR Interpreting Telephony Research Laboratories, 1991.

[98] TUCKER, R. C. F., CAREY, M. J., AND PARRIS, E. S. Automatic language identification using sub-word models. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Adelaide, Australia, April 1994), vol. 1, pp. 301 – 304.

[99] UEBERLA, J. P. More efficient clustering of n-grams for statistical language modeling. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 1257–1260.

[100] VILAR, J. M., VIDAL, E., AND MARZAL, A. Learning language translation in limited domains using finite-state models: some extensions and improvements. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 1231–1230.

[101] WRIGHT, J., CAREY, M., AND PARRIS, E. Statistical models for topic identification using phoneme substrings. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Atlanta, USA, May 1996), vol. 1, pp. 307–310.

[102] YAN, Y. *Development of An Approach to Language Identification based on Language-dependent Phone Recognition.* PhD thesis, Oregon Graduate Institute, October 1995.

[103] YAN, Y., AND BARNARD, E. An approach to automatic language identification based on language-dependent phone recognition. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Detroit,USA, May 1995), vol. 5, pp. 3511–3514.

[104] YAN, Y., AND BARNARD, E. An approach to language identification with enhanced language model. In *Proceedings Eurospeech* (Madrid, Spain, September 1995), vol. 2, pp. 1351–1354.

[105] YAN, Y., AND BARNARD, E. Neural networks and linear classifiers. automatic language identification. In *In Proceedings IEEE International Conference on Neural Networks and Signal Processing* (December 1995), pp. 812–815.

[106] YAN, Y., AND BARNARD, E. Experiments for an approach to langugage identivication with conversational telephone speech. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Atlanta, USA, May 1996), vol. 1, pp. 789–792.

[107] YAN, Y., BARNARD, E., AND COLE, R. Development of an approach to automatic language identification based on phone recognition. *Computer Speech and Language 10*, 1 (January 1996), 37–53.

[108] YAN, Y., BERKLING, K. M., AND BARNARD, E. Bigram models and phoneme clusters for language identification. In *Proceedings Speech Research Symposium XIV SRS* (Baltimore, Maryland, June 1994), pp. 22–30.

[109] ZIEGLER, D.-V. *The Automatic Identification of Languages Using Linguistic Recognition Signals.* PhD thesis, State University of New York at Buffalo, June 1991.

[110] ZISSMAN, M., GLEASON, T., REKART, D., AND LOSIEVICZ, B. Automatic dialect identification of extemporaneous, conversational Latin American Spanish speech. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Atlanta, USA, May 1996), vol. 1, pp. 777–780.

[111] ZISSMAN, M. A. Automatic language identification using gaussian mixtures and hidden markov models. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (April 1993), vol. 2, pp. 399–402.

[112] ZISSMAN, M. A. Language identification using phoneme recognition and phonotactic language modeling. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (May 1995), vol. 5, pp. 3503–3506.

[113] ZISSMAN, M. A., AND SINGER, E. Automatic language identification of telephone speech messages using phoneme recognition and n-gram modelling. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing* (Adelaide, Australia, April 1994), vol. 1, pp. 305–308.

Appendix A

Labeling Conventions

IPA, Worldbet, and OGIbet English Broad Phonetic Labels

Center for Spoken Language Understanding - Oregon Graduate Institute of Science & Technology

IPA	Worldbet	OGIbet	Example	Category
iː	i:	iy	beet	
ɪ	I	ih	bit	Front
e	E	eh	bet	Vowels
æ	@	ae	bat	
ɨ	I_x	ix	roses	
ʉ	u_x	ux	suit	Central
ə	^	ax	above	Vowels
ɚ	^_O		to go	
ɒ	S		pot	(British)
uː	u	uw	boot	
ʊ	U	uh	book	Back
ʌ	^	ah	above	Vowels
ɔ	>	ao	caught	
ɑ	A	aa	father	
ɝ	3r	er	bird	Retro-
ɚ	^r	axr	butter	flexes
eɪ	ei	ey	bay	
aɪ	aI	ay	bye	Diph-
ɔɪ	>i	oy	boy	thongs
iʊ	iU		few	
aʊ	aU	aw	about	
oʊ	oU	ow	boat	
iə	i^		here	(British)
eə	e^		there	(British)
uə	u^		poor	(British)
pʰ	ph	p	pan	Voiceless
tʰ	th	t	tan	Plosives
kʰ	kh	k	can	
b	b	b	ban	Voiced
d	d	d	dan	Plosives
g	g	g	gander	
m	m	m	me	
n	n	n	knee	Nasals
ŋ	N	ng	sing	
ɾt	th_(	dx	writer	Flaps
ɾd	d_(	dx	rider	
f	f	f	fine	
θ	T	th	thigh	Voiceless
s	s	s	sign	Fricatives
ʃ	S	sh	assure	
h	h	hh	hope	
v	v	v	vine	
ð	D	dh	thy	Voiced
z	z	z	resign	Fricatives
ʒ	Z	zh	azure	
ʧ	tS	ch	church	Affricates
ʤ	dZ	jh	judge	
l	l	l	lent	Glides
ɹ	9r	r	rent	
j	j	y	yes	(approxi-
w	w	w	went	mants)
m̩	m=	em	bottom	
n̩	n=	en	button	Syllabics
ŋ̩	N=	eng		
l̩	l=	el	bottle	

IPA	Worldbet	OGIbet	Example	Category
	pc	pcl	_pan	Voiceless
	tc	tcl	_tan	Plosive
	kc	kcl	_can	Closures
	bc	bcl	_ban	Voiced
	dc	dcl	_dan	Plosive
	gc	gcl	_gander	Closures
	tSc	chcl	_church	Affricate
	dZc	jhcl	_judge	Closures
	+	.epi	epinthetic closure	

IPA	Worldbet	OGIbet	Type of Diacritic
tʰ	_h	-h	aspirated
	_x		centralized
t̪ d̪	_[		dental
	_(		flapped (consonant)
	_F		fricated stop
	_?*	q	glottal onset
t̰	_?	-q	glottalized
dˡ	_l		lateral release
iː	_:	-el	lengthened
dⁿ	_n		nasal release
ẽ	_~	-n	nasalized
	_NL	.nitl	not in the language
tʲ	_j		palatalized
ɚ	_r	-r	retroflexion
ɨ	_i		less rounded
ʉ	_w		more rounded
s̬	_v		voiced
n̥ d̥	_0		voiceless
	_*	-	waveform cut off
Worldbet, as modified at OGI			
	_fp	-fp	filled pause
	_ln	-ln	line noise corruption
	_bn		background noise

Worldbet	OGIbet	Non Speech Sound Item
.bn	.bn	background noise
.br	.br	breath noise
.cough	.cough	cough
.ct	.ct	clear throat
.laugh	.laugh	laugh
.ln	.ln	lin noise
.ls	.ls	lip smack
.ns	.ns	human, not speech
.sneeze	.sneeze	sneeze
.tc	.tc	tongue click
Worldbet, as modified at OGI		
.beep	.beep	beep
.burp	.burp	burp
.fp	.fp	filled pause
.pau	.pau	pause or silence
.sniff	.sniff	sniff
.uu	.unk	unintelligible speech
.vs	.vs	squeak, voice crack
.glot	glot	glottalization

Figure A.1: Table of Worldbet symbols.

Appendix B

Label Statistics

Table B.1: Phoneme Frequencies in Labeled files

LABEL	Hindi	German	English	Spanish	Japanese	Mandarin
&	0.0061	0.0164	0.0280	0.0020	0.0003	0.0206
&r	0.0000	0.0000	0.0041	0.0000	0.0000	0.0014
.pau	0.1155	0.0853	0.0911	0.0924	0.0981	0.0912
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0434
3	0.0000	0.0000	0.0000	0.0098	0.0000	0.0000
3r	0.0000	0.0000	0.0147	0.0000	0.0000	0.0000
4r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0162
9r	0.0000	0.0000	0.0283	0.0000	0.0000	0.0000
>	0.0154	0.0172	0.0087	0.0000	0.0000	0.0290
> Y	0.0000	0.0041	0.0012	0.0000	0.0000	0.0000
?	0.0009	0.0002	0.0004	0.0019	0.0024	0.0003
?*	0.0021	0.0074	0.0122	0.0041	0.0073	0.0006
@	0.0158	0.0062	0.0200	0.0001	0.0000	0.0049
A	0.0000	0.0000	0.0164	0.0000	0.0000	0.0463
A:	0.0000	0.0200	0.0000	0.0000	0.0000	0.0000
C	0.0000	0.0230	0.0000	0.0000	0.0000	0.0000
D	0.0000	0.0000	0.0192	0.0175	0.0000	0.0000
E	0.0424	0.0747	0.0237	0.0290	0.0000	0.0267
Eax	0.0000	0.0110	0.0000	0.0000	0.0000	0.0000
I	0.0297	0.0490	0.0501	0.0044	0.0000	0.0128
Ix	0.0027	0.0020	0.0140	0.0000	0.0000	0.0000
K	0.0000	0.0197	0.0000	0.0050	0.0000	0.0000
N	0.0022	0.0041	0.0087	0.0057	0.0004	0.0384
S	0.0057	0.0143	0.0059	0.0010	0.0183	0.0186
T	0.0000	0.0000	0.0057	0.0002	0.0000	0.0000
U	0.0167	0.0200	0.0028	0.0009	0.0000	0.0000
V	0.0000	0.0000	0.0000	0.0146	0.0000	0.0000
up	0.0814	0.0000	0.0527	0.0000	0.0000	0.0000
a	0.0627	0.0378	0.0000	0.0953	0.1201	0.0000
aI	0.0000	0.0000	0.0180	0.0015	0.0002	0.0000
aU	0.0004	0.0069	0.0057	0.0000	0.0000	0.0132
ai	0.0035	0.0200	0.0000	0.0000	0.0000	0.0192
b	0.0185	0.0181	0.0132	0.0064	0.0066	0.0000
bc	0.0191	0.0153	0.0120	0.0043	0.0058	0.0000
cC	0.0000	0.0000	0.0000	0.0000	0.0000	0.0161
cCc	0.0000	0.0000	0.0000	0.0000	0.0000	0.0188
cCh	0.0000	0.0000	0.0000	0.0000	0.0000	0.0093
d	0.0041	0.0358	0.0198	0.0000	0.0000	0.0000
dZ	0.0121	0.0002	0.0051	0.0011	0.0065	0.0000
dc	0.0009	0.0267	0.0277	0.0000	0.0000	0.0000
dcc	0.0118	0.0000	0.0000	0.0135	0.0211	0.0000
dccc	0.0104	0.0000	0.0000	0.0081	0.0266	0.0000
doo	0.0000	0.0000	0.0106	0.0000	0.0000	0.0000
drc	0.0099	0.0000	0.0000	0.0000	0.0000	0.0000

Table B.2: Phoneme Frequencies in Labeled files

LABEL	Hindi	German	English	Spanish	Japanese	Mandarin
e	0.0363	0.0000	0.0000	0.0899	0.0684	0.0000
e:	0.0000	0.0086	0.0000	0.0000	0.0004	0.0000
ei	0.0000	0.0000	0.0152	0.0000	0.0000	0.0149
f	0.0045	0.0205	0.0155	0.0043	0.0018	0.0048
g	0.0097	0.0184	0.0080	0.0034	0.0131	0.0000
gc	0.0081	0.0153	0.0061	0.0022	0.0085	0.0000
h	0.0189	0.0119	0.0100	0.0001	0.0088	0.0219
hs	0.0000	0.0000	0.0000	0.0085	0.0000	0.0000
i	0.0292	0.0000	0.0000	0.0574	0.0656	0.0000
i:	0.0000	0.0085	0.0336	0.0000	0.0000	0.0305
j	0.0146	0.0042	0.0067	0.0028	0.0150	0.0608
k	0.0369	0.0000	0.0000	0.0274	0.0503	0.0186
kH	0.0039	0.0000	0.0000	0.0000	0.0000	0.0048
kc	0.0408	0.0109	0.0258	0.0273	0.0467	0.0183
kh	0.0000	0.0110	0.0232	0.0000	0.0000	0.0000
l	0.0228	0.0291	0.0308	0.0346	0.0003	0.0147
m	0.0365	0.0305	0.0251	0.0360	0.0296	0.0139
n	0.0303	0.0891	0.0586	0.0501	0.0650	0.0565
nj	0.0000	0.0000	0.0001	0.0032	0.0000	0.0000
o	0.0201	0.0014	0.0000	0.0713	0.1007	0.0000
o:	0.0000	0.0059	0.0000	0.0000	0.0001	0.0000
oU	0.0000	0.0000	0.0118	0.0000	0.0000	0.0353
oax	0.0000	0.0027	0.0000	0.0000	0.0000	0.0000
p	0.0137	0.0000	0.0000	0.0185	0.0025	0.0129
pc	0.0124	0.0056	0.0133	0.0166	0.0021	0.0119
ph	0.0000	0.0053	0.0133	0.0000	0.0000	0.0000
roo	0.0034	0.0000	0.0000	0.0385	0.0284	0.0000
rr	0.0335	0.0028	0.0000	0.0025	0.0000	0.0055
s	0.0272	0.0432	0.0426	0.0612	0.0248	0.0044
sr	0.0000	0.0000	0.0000	0.0000	0.0000	0.0246
tS	0.0059	0.0005	0.0042	0.0034	0.0066	0.0021
tc	0.0000	0.0500	0.0423	0.0000	0.0000	0.0000
tcc	0.0242	0.0000	0.0000	0.0432	0.0436	0.0349
tccH	0.0047	0.0000	0.0000	0.0000	0.0042	0.0094
tccc	0.0271	0.0000	0.0000	0.0414	0.0471	0.0365
tccc:	0.0001	0.0000	0.0000	0.0000	0.0057	0.0000
th	0.0038	0.0404	0.0325	0.0024	0.0000	0.0000
trc	0.0141	0.0000	0.0000	0.0000	0.0000	0.0000
ts	0.0065	0.0079	0.0000	0.0000	0.0000	0.0179
tsR	0.0000	0.0000	0.0000	0.0000	0.0000	0.0055
tsc	0.0000	0.0000	0.0000	0.0000	0.0000	0.0147
tsr	0.0000	0.0000	0.0000	0.0000	0.0000	0.0127
tsrc	0.0000	0.0000	0.0000	0.0000	0.0000	0.0132
u	0.0000	0.0012	0.0000	0.0249	0.0360	0.0149
u:	0.0053	0.0041	0.0102	0.0000	0.0000	0.0000
v	0.0000	0.0167	0.0145	0.0000	0.0000	0.0001
w	0.0122	0.0000	0.0193	0.0040	0.0109	0.0437
x	0.0000	0.0000	0.0000	0.0062	0.0000	0.0000
y	0.0000	0.0065	0.0000	0.0000	0.0000	0.0129
yax	0.0000	0.0052	0.0000	0.0000	0.0000	0.0000
z	0.0031	0.0073	0.0177	0.0000	0.0000	0.0000

Table B.3: Phoneme Frequencies in Aligned files

LABEL	Hindi	German	English	Spanish	Japanese	Mandarin
&	0.0029	0.0001	0.0000	0.0001	0.0039	0.0035
&r	0.0015	0.0000	0.0000	0.0000	0.0019	0.0010
.pau	0.0539	0.0418	0.0388	0.0424	0.0513	0.0622
2	0.0012	0.0000	0.0000	0.0000	0.0016	0.0044
3	0.0041	0.0019	0.0037	0.0051	0.0046	0.0025
3r	0.0013	0.0000	0.0000	0.0000	0.0021	0.0025
4r	0.0008	0.0000	0.0000	0.0000	0.0032	0.0068
9r	0.0031	0.0000	0.0005	0.0000	0.0022	0.0007
>	0.0208	0.0264	0.0301	0.0217	0.0177	0.0239
$>Y$	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
?*	0.0108	0.0102	0.0100	0.0087	0.0121	0.0132
?	0.0051	0.0025	0.0023	0.0027	0.0054	0.0079
@	0.0183	0.0155	0.0235	0.0178	0.0173	0.0196
A	0.0049	0.0000	0.0000	0.0000	0.0018	0.0065
A:	0.0111	0.0142	0.0098	0.0081	0.0100	0.0152
C	0.0015	0.0057	0.0009	0.0012	0.0031	0.0023
D	0.0016	0.0000	0.0000	0.0000	0.0024	0.0010
E	0.0229	0.0308	0.0330	0.0292	0.0229	0.0252
Eax	0.0007	0.0000	0.0000	0.0000	0.0007	0.0004
I	0.0122	0.0127	0.0189	0.0094	0.0132	0.0075
Ix	0.0031	0.0001	0.0001	0.0000	0.0032	0.0019
K	0.0023	0.0019	0.0004	0.0003	0.0038	0.0038
N	0.0046	0.0039	0.0040	0.0030	0.0044	0.0053
S	0.0019	0.0031	0.0027	0.0020	0.0044	0.0072
T	0.0005	0.0000	0.0000	0.0000	0.0005	0.0005
U	0.0216	0.0286	0.0174	0.0226	0.0150	0.0166
V	0.0055	0.0014	0.0019	0.0057	0.0022	0.0016
up	0.0106	0.0052	0.0054	0.0066	0.0077	0.0061
a	0.0292	0.0298	0.0234	0.0373	0.0371	0.0203
aI	0.0084	0.0180	0.0257	0.0096	0.0121	0.0151
aU	0.0056	0.0056	0.0081	0.0025	0.0064	0.0112
ai	0.0041	0.0000	0.0000	0.0000	0.0014	0.0051
b	0.0131	0.0033	0.0065	0.0028	0.0047	0.0046
bc	0.0118	0.0034	0.0067	0.0027	0.0045	0.0042
cC	0.0104	0.0057	0.0045	0.0033	0.0109	0.0208
cCc	0.0095	0.0054	0.0047	0.0030	0.0116	0.0238
cCh	0.0007	0.0003	0.0006	0.0002	0.0017	0.0068
d	0.0031	0.0105	0.0108	0.0050	0.0049	0.0054
dZ	0.0023	0.0004	0.0011	0.0004	0.0021	0.0016
dc	0.0061	0.0127	0.0133	0.0068	0.0068	0.0092
dcc	0.0129	0.0109	0.0078	0.0134	0.0110	0.0072
dccc	0.0117	0.0082	0.0066	0.0118	0.0106	0.0060
doo	0.0089	0.0068	0.0155	0.0159	0.0109	0.0057
drc	0.0153	0.0124	0.0134	0.0100	0.0079	0.0107

Table B.4: Phoneme Frequencies in Aligned files

LABEL	Hindi	German	English	Spanish	Japanese	Mandarin
e	0.0495	0.0361	0.0386	0.0681	0.0355	0.0212
e:	0.0028	0.0044	0.0025	0.0036	0.0024	0.0022
ei	0.0029	0.0000	0.0000	0.0000	0.0027	0.0027
f	0.0036	0.0000	0.0000	0.0000	0.0035	0.0016
g	0.0136	0.0133	0.0134	0.0122	0.0094	0.0098
gc	0.0009	0.0000	0.0000	0.0000	0.0013	0.0007
h	0.0161	0.0205	0.0185	0.0135	0.0144	0.0310
hs	0.0133	0.0104	0.0082	0.0224	0.0101	0.0056
i	0.0200	0.0102	0.0066	0.0200	0.0110	0.0070
i:	0.0264	0.0341	0.0396	0.0279	0.0301	0.0422
j	0.0210	0.0140	0.0140	0.0236	0.0215	0.0326
k	0.0459	0.0533	0.0461	0.0607	0.0541	0.0241
kH	0.0097	0.0166	0.0225	0.0074	0.0106	0.0132
kc	0.0597	0.0697	0.0706	0.0673	0.0673	0.0382
kh	0.0023	0.0000	0.0000	0.0000	0.0025	0.0023
l	0.0165	0.0136	0.0133	0.0200	0.0133	0.0118
m	0.0116	0.0026	0.0024	0.0044	0.0072	0.0048
n	0.0567	0.0962	0.0696	0.0713	0.0525	0.0732
nj	0.0010	0.0000	0.0000	0.0000	0.0003	0.0000
o	0.0245	0.0225	0.0176	0.0327	0.0338	0.0172
o:	0.0011	0.0010	0.0001	0.0005	0.0019	0.0014
oU	0.0076	0.0078	0.0104	0.0091	0.0131	0.0228
oax	0.0004	0.0000	0.0000	0.0000	0.0008	0.0005
p	0.0230	0.0193	0.0150	0.0273	0.0205	0.0153
pc	0.0426	0.0459	0.0409	0.0490	0.0422	0.0370
ph	0.0101	0.0151	0.0166	0.0105	0.0100	0.0109
roo	0.0008	0.0000	0.0000	0.0000	0.0019	0.0004
rr	0.0168	0.0108	0.0143	0.0186	0.0086	0.0099
s	0.0339	0.0669	0.0678	0.0624	0.0530	0.0341
sr	0.0059	0.0108	0.0124	0.0079	0.0074	0.0259
tS	0.0045	0.0064	0.0108	0.0038	0.0044	0.0048
tc	0.0042	0.0018	0.0022	0.0004	0.0091	0.0053
tcc	0.0079	0.0011	0.0005	0.0024	0.0086	0.0031
tccH	0.0021	0.0035	0.0037	0.0014	0.0025	0.0016
tccc	0.0080	0.0012	0.0010	0.0025	0.0078	0.0033
tccc:	0.0006	0.0004	0.0001	0.0002	0.0012	0.0007
th	0.0009	0.0000	0.0000	0.0000	0.0032	0.0015
trc	0.0080	0.0082	0.0101	0.0044	0.0053	0.0073
ts	0.0046	0.0051	0.0028	0.0019	0.0064	0.0089
tsR	0.0006	0.0000	0.0000	0.0000	0.0005	0.0024
tsc	0.0012	0.0000	0.0000	0.0000	0.0021	0.0034
tsr	0.0021	0.0015	0.0020	0.0010	0.0021	0.0069
tsrc	0.0007	0.0000	0.0000	0.0000	0.0011	0.0042
u	0.0028	0.0000	0.0000	0.0000	0.0049	0.0030
u:	0.0019	0.0000	0.0000	0.0000	0.0013	0.0010
v	0.0064	0.0150	0.0172	0.0090	0.0117	0.0066
w	0.0127	0.0162	0.0295	0.0153	0.0138	0.0197
x	0.0041	0.0018	0.0025	0.0039	0.0022	0.0012
y	0.0021	0.0021	0.0022	0.0016	0.0028	0.0059
yax	0.0007	0.0000	0.0000	0.0000	0.0002	0.0001
z	0.0015	0.0011	0.0021	0.0009	0.0025	0.0025

Table B.5: Phoneme Frequencies after clustering to 59 phonemes (aligned)

LABEL	Hindi	German	English	Spanish	Japanese	Mandarin
&	0.0876	0.0164	0.1276	0.0119	0.0003	0.0654
.pau	0.1155	0.0853	0.0911	0.0924	0.0981	0.0912
4r	0.0000	0.0065	0.0000	0.0000	0.0000	0.0291
>	0.0154	0.0199	0.0087	0.0000	0.0000	0.0290
> Y	0.0000	0.0100	0.0012	0.0000	0.0001	0.0000
?	0.0072	0.0433	0.0324	0.0060	0.0097	0.0009
@	0.0158	0.0171	0.0200	0.0001	0.0000	0.0049
A	0.0000	0.0200	0.0164	0.0000	0.0000	0.0463
C	0.0000	0.0230	0.0000	0.0000	0.0000	0.0000
D	0.0053	0.0053	0.0294	0.0569	0.0361	0.0149
E	0.0424	0.0747	0.0237	0.0290	0.0000	0.0267
I	0.0297	0.0490	0.0501	0.0044	0.0000	0.0128
Ix	0.0027	0.0020	0.0140	0.0000	0.0000	0.0000
K	0.0000	0.0197	0.0000	0.0050	0.0000	0.0000
N	0.0387	0.0346	0.0339	0.0416	0.0300	0.0523
S	0.0246	0.0263	0.0159	0.0012	0.0271	0.0652
T	0.0045	0.0205	0.0212	0.0107	0.0018	0.0048
U	0.0167	0.0200	0.0028	0.0009	0.0000	0.0000
a	0.0627	0.0378	0.0000	0.0953	0.1201	0.0000
aI	0.0035	0.0200	0.0180	0.0015	0.0002	0.0192
aU	0.0004	0.0069	0.0057	0.0000	0.0000	0.0132
b	0.0185	0.0181	0.0132	0.0064	0.0066	0.0000
bc	0.0191	0.0153	0.0120	0.0043	0.0058	0.0000
cC	0.0000	0.0000	0.0000	0.0000	0.0000	0.0161
cCc	0.0000	0.0000	0.0000	0.0000	0.0000	0.0188
cCh	0.0121	0.0002	0.0051	0.0011	0.0065	0.0093
dc	0.0009	0.0267	0.0277	0.0000	0.0000	0.0000
dcc	0.0118	0.0000	0.0000	0.0135	0.0211	0.0000
dccc	0.0284	0.0153	0.0061	0.0103	0.0351	0.0000
doo	0.0034	0.0000	0.0106	0.0385	0.0284	0.0000
e	0.0363	0.0000	0.0000	0.0899	0.0684	0.0000
e:	0.0000	0.0086	0.0152	0.0000	0.0004	0.0149
g	0.0097	0.0184	0.0080	0.0034	0.0131	0.0000
hs	0.0000	0.0000	0.0000	0.0085	0.0000	0.0000
i	0.0292	0.0000	0.0000	0.0574	0.0656	0.0000
i:	0.0000	0.0085	0.0336	0.0000	0.0000	0.0305
j	0.0146	0.0094	0.0067	0.0028	0.0150	0.0608
k	0.0369	0.0000	0.0000	0.0274	0.0503	0.0186
kH	0.0039	0.0110	0.0232	0.0000	0.0000	0.0048
kc	0.0408	0.0109	0.0258	0.0273	0.0467	0.0183
l	0.0228	0.0291	0.0308	0.0346	0.0003	0.0147
n	0.0303	0.0891	0.0587	0.0533	0.0650	0.0565
o	0.0201	0.0014	0.0000	0.0713	0.1007	0.0000
oU	0.0000	0.0000	0.0118	0.0000	0.0000	0.0353
p	0.0378	0.0000	0.0000	0.0617	0.0461	0.0478
pc	0.0124	0.0056	0.0133	0.0166	0.0021	0.0119
ph	0.0000	0.0053	0.0133	0.0000	0.0000	0.0000
rr	0.0335	0.0028	0.0000	0.0025	0.0000	0.0055
s	0.0272	0.0432	0.0426	0.0612	0.0248	0.0044
tS	0.0059	0.0005	0.0042	0.0034	0.0066	0.0076
tc	0.0001	0.0500	0.0423	0.0000	0.0057	0.0147
tccH	0.0085	0.0404	0.0325	0.0024	0.0043	0.0094
tccc	0.0271	0.0000	0.0000	0.0414	0.0471	0.0365
trc	0.0141	0.0000	0.0000	0.0000	0.0000	0.0132
ts	0.0065	0.0079	0.0000	0.0000	0.0000	0.0179
tsr	0.0000	0.0000	0.0000	0.0000	0.0000	0.0127
v	0.0000	0.0167	0.0145	0.0000	0.0000	0.0001
w	0.0122	0.0000	0.0193	0.0040	0.0109	0.0437
z	0.0031	0.0073	0.0177	0.0000	0.0000	0.0000

Appendix C

English vs. German

C.1 Clustering Trees

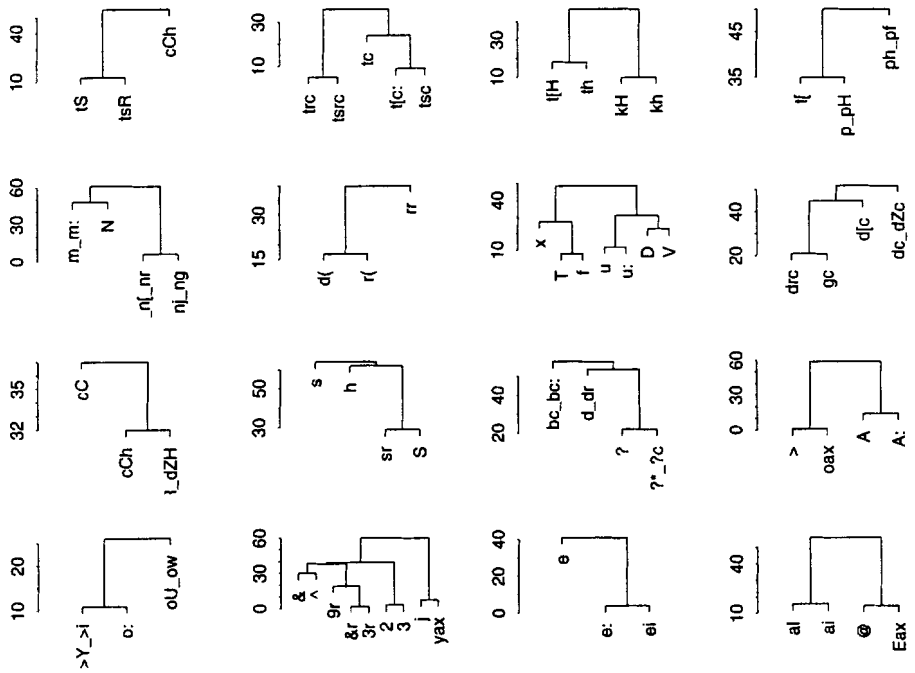

Figure C.1: Clustering of phonemes. The last shown merge represents the forbidden merge.

C.2 Merged Classes

Table C.1: Table of merged classes

& ˆ &r 3r 9r 2 3	.pau	4r y Y y:
> oax	> *Y* > *i* o:	? ?* ?c d dr
@ Eax	A A:	C
D V u uax u:	E 8	I If
Ix	K G	N m m:
S sr h	T f x	U
a	aI ai	aU
b bH	bc bc:	cCh dZ dR dZH
cC	cCc	d(r(r(H
dc dZc	d[d[: d[H d[z	d[c drc gc
e	e: ei	g gH
hs	i ih	i:
j yax	k	kH kh
kc kc:	l L l: l(	n n: n[nr nj ng
o 7	oU owl	p pH t[
pc pc:	ph pf	rr r r+
s s:	tS tsH tsR	tc t[c: tsc tSc
t[H t[s th t tR tSH	t[c t[sc	trc trc: tsrc
ts tr	tsr	v
w	z Z	

C.3 Disallowed Merges

Table C.2: Disallowed merges for single phonemes.

i,ih	E,8
a	aU
I,If	Ix
l,L,l:,l(	w
U	o,7
C	z
K,G	hs
b,bH	v
d[,d[:,d[H,d[z	g,gH
ts	tsr
k	kc,kc:
pc,pc:	t[c

Table C.3: Disallowed merges for phoneme classes.

aI,ai	@,Eax
e	e:,ei
j,yax	&, ˆ,9r,&r,3r,2,3
4r,y	i:
bc,bc:	d,dr,?,?*,?c
>,oax	A,A:
> Y,> i,o:	oU,ow
cC	cCh,dZ,dR,dZH
d(,r(	rr
tS,tsR	cCh
s,T,f	u,u:,D,V
trc,tsrc	tc,t[c:,tsc
sr,S,h	s
kH,kh	t[H,th
drc,gc,d[c	dc,dZc
m,m:,N	n,n:,n[,nr,nj,ng
t[,p,pH	ph,pf

C.4 Word list to discriminate EN vs. GE

Table C.4: The List of Words discriminating English vs. German.

Word List	**Example**	**Language(α)**	error
doo	rider	EN(1.000000)	0.417390
U n	und	GE(-0.628283)	0.433311
n	und	GE(-0.132326)	0.437518
a	aber	GE(-0.262205)	0.440138
Ix	ich	EN(0.293439)	0.440718
tc ts	zu	GE(-0.659480)	0.441602
& w	wow	EN(1.780642)	0.443074
ts U n	zunehmen	GE(-2.005834)	0.448822
e: doo	ladder	EN(2.173229)	0.450145
.pau ? I C	..ich	GE(-2.676625)	0.450492
w &	where	EN(1.014912)	0.450633
& e:	and	EN(2.099001)	0.451763
C 4r	Becher	GE(-2.768908)	0.451810
U	Und	GE(-0.251789)	0.452168
& dc	led	EN(1.296263)	0.452391
a n	and	GE(-0.510496)	0.452703
? U	..Und	GE(-2.001755)	0.454155
C tc tccH E	Geschichte	GE(-3.997612)	0.454308
ts	zu	GE(-0.389870)	0.454719
g E S	Geschichte	GE(-2.621442)	0.455105
n a	nain	GE(-2.607831)	0.455241
A K	mach	GE(-1.355402)	0.456363
K	mach	GE(-0.251271)	0.456574
pc ph e:	family	EN(3.735772)	0.456640
ts E n	verletzen	GE(-4.165718)	0.456749
e: C tc tccH	echt	GE(-4.952376)	0.457405
C tc tccH	nicht	GE(-1.017309)	0.457442
& bc b &	about	EN(2.625859)	0.457516
tS	check	EN(0.548847)	0.457538
tc ts U n	zunehmen	GE(-2.685595)	0.457641
tc tS &	check	EN(1.631012)	0.457651
TOTAL ESTIMATED			0.019

Appendix D

Results using Neural Network Implementation

Correct Language	Classified Language					
	59 classes		95 classes		59 classes with weighting	
	EN	GE	EN	GE	EN	GE
EN	18	1	17	2	17	2
GE	2	18	4	16	5	15

Table D.1: Number of files classified from test set.

Classifier	Misclassified Files
59 classes	German call 90 German call 94 English call 63
59 classes with alpha	German call 79 German call 81 German call 90 German call 93 German call 94 English call 63 English call 72
95 classes	German call 79 German call 88 German call 90 German call 94 English call 72 English call 81

Table D.2: Names of misclassified files from test set.

Appendix E

Inexact Sequence Matching

In this appendix we will discuss an extension to the algorithm implemented in Chapter 5. The aim is to build a system which is more robust and therefore generalizes better to previously unseen test data. By extending the model we want to improve generalization to test data in the face of factors such as pronunciation or dialect variability within a language, which gives rise to inaccurate alignment. In order to achieve this goal, sequences are matched in an inexact manner. In the following sections, we will motivate this extension and describe the algorithm. Finally, we compare the results to the baseline system including only exact sequence matching. The algorithm studied in this appendix is a practical realisation of the ideas introduced in Chapter 4 and the hope is that it will be more successful in the practical setting than predicted by theory.

E.1 Motivation

In Section 5.5.3 we found that even a hard task such as the distinction between English and German can be accomplished perfectly with our techniques if no misrecognitions occur. In an attempt to approximate that result with limited recognition accuracy, we try to recover the occurrences of our target strings ("words") even when they are computed by misrecognitions. To do this, a "Word" is created by associating a set of sequences with each other by allowing for the possibility of misrecognition. The aim is to associate

sequences which cover the variability of a word within one language without sacrificing any discriminability across languages. The idea is illustrated with a simple example in Figure E.1. Shown are the phonemic representations of three speakers in German and English saying the same sentence in their respective languages. Notice that the boxes contain the same word pronounced differently by each speaker.

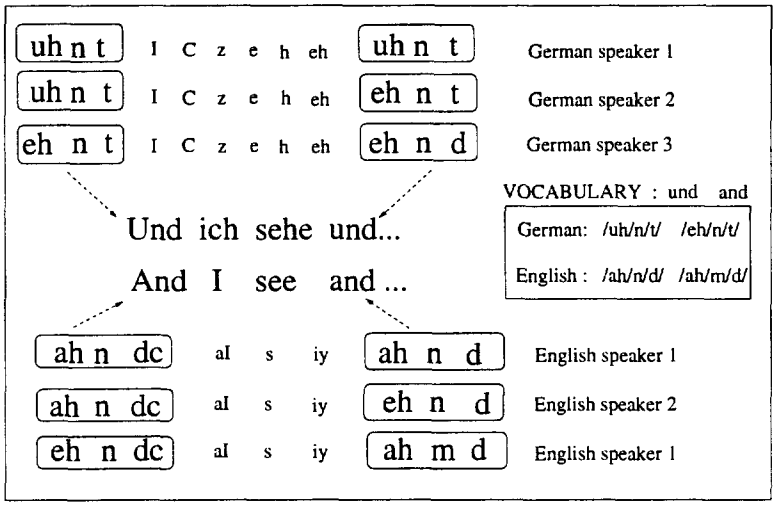

Figure E.1: Example of speaker dependent pronunciations of same word.

Either sequence **/uh/n/tcl/t** or **/eh/n/tcl/t** by itself is not sufficient to describe the language German. Treating both sequences as separate features is inadequate because a non-occurrence of one sequence may be misinterpreted to mean that the incoming utterance is not German. However, both sequences together describe the word "und" and discriminate German from English. The same is true for sequences **/ah/n/dcl/d/** and **/ah/m/dcl/d/** for English. The list of sequences can be thought of as an OR function where each of the sequences within the "word" they define is not expected to occur in all utterances. However, ideally, the word as a whole occurs with uniform frequency in each file. Uniting both sequences to represent the words "und" and "and" respectively results in a mean occurrence of twice per file for the first two utterances and zero occurrence in the other language, resulting in complete discrimination.

But how much inaccuracy can be allowed before language discrimination is sacrificed by an overlap of pronunciations as depicted in Figure E.2? Adding **/eh/n/dcl/d/** to the list of allowed pronunciations for German, may improve the recognition for German but not the discrimination from English because we now have created an overlap in pronunciations. In rapid speech one often mispronounces or partially pronounces a word. In this example the words "and" in English and "und" in German come very close to each other. Using real data, such overlapping pronunciations are prominent especially for shorter sequences. By modeling the occurrence frequencies of sequences with normal distributions, the discrimination error between languages can be estimated. The algorithm discussed in this chapter will use the error prediction method developed in Chapter 3 and applied in Chapter 5 to allow inaccurate string matchings while limiting the degree of variability when the discrimination between two languages is sacrificed.

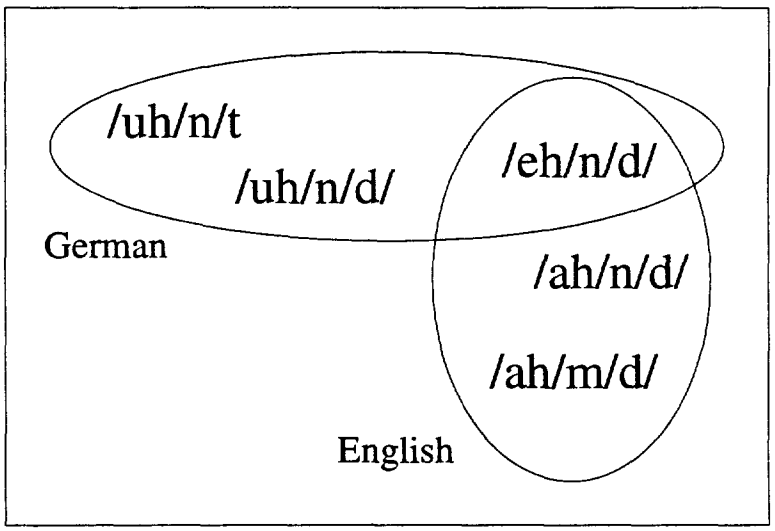

Figure E.2: Example of pronunciations overlapping languages.

E.2 Terminology

In order to specify sequences in an inexact manner we need to find a measure of allowed inaccuracy. If we imagine the space of all sequences partitioned as shown in Fig. E.3 then we define the following terms:

<u>Center</u> The center sequence of a set of associated sequences, ie. their representative sequence

<u>Radius</u> The **Radius** defines the degree of allowed inaccuracy between a sequence and a **Center**.

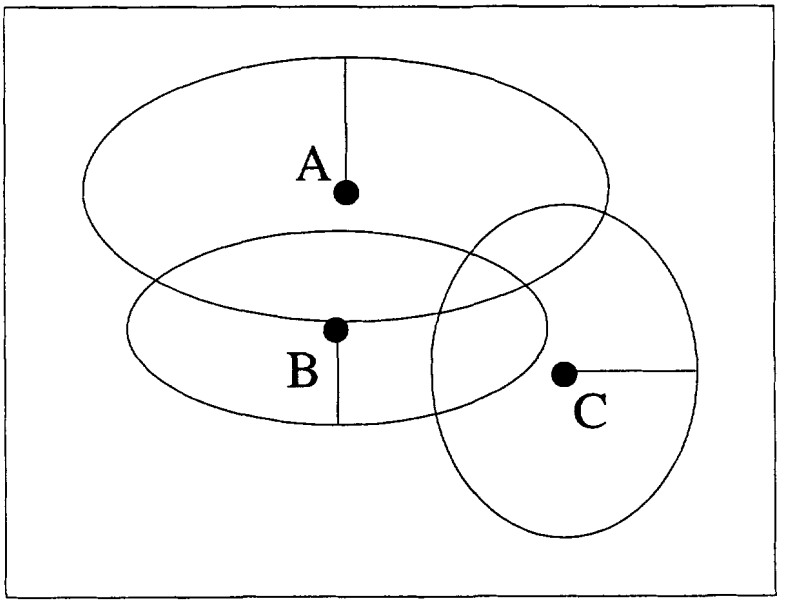

Figure E.3: Space of All Sequences. A,B, and C represent the centers of the three sets. Each set is associated with a radius shown by the line. Sets may overlap.

The issue then becomes one of associating a set of sequences with each other, finding the appropriate **Center** to represent this set and defining a corresponding **Radius**. An optimal list of "words" is then derived similar to the list of words in the previous chapter by sorting them according to the estimated corresponding error and selecting the N most discriminating "words". Language identification is then performed by matching the predetermined set of **Center**s within the corresponding **Radius** and using the frequency counts as features.

There are some research questions to be answered with respect to forming sets of sequences. We address these in three parts in the next sections. After describing the distance measure between sequences, I will discuss the weighting of each sequence within a word. The final important issue is the ordering in which the sequences are added to a word, because it will determine the number of sequences the final word encompasses.

E.3 Distance Function

Two sequences are associated with each other by a distance score. Distance scores between two sequences are generally calculated using dynamic time warping (dtw). We will approach the derivation of the score in a two-step process. The first dtw-algorithm is frame based. The second one is segment based and builds on the first algorithm.

The goal is to match two strings of tokens, both corresponding to the same utterance phonemically labeled by hand. The corresponding utterance has also been automatically aligned using the process described in Section 5.2. This alignment forms the string to be matched against the labeled data. Sequences are matched using a frame-based confusion matrix derived from the labeled and aligned files. Each entry $[a, l]$ in the matrix corresponds to $p(a|l)$, denoting the probability that frame **l** is classified as **a** by the neural-network phoneme recognizer described in Section 5.2.1. The cost **c** of a substitution of

label **a** for label **l** is derived from $p(a|l)$.

$$c[l \rightarrow a] = -log[p(a|l)] \tag{E.1}$$

The distance score between the input sequence **A** of length j and a template sequence **L** of length i to be matched is then calculated using cost **c** according to the general principles of dynamic programming:

$$C[L_i, A_j] = max \begin{cases} C[L_{i-1}, A_{j-1}] & + & c(L_i \rightarrow A_j) \\ C[L_{i-1}, A_j] & + & c(L_i \rightarrow A_j) \\ C[L_i, A_{j-1}] & + & c(L_i \rightarrow A_j) \end{cases} \tag{E.2}$$

The score C in Equation E.2 relates the probability of matching an aligned sequence **A** to a labeled sequence **L** by taking the inverse of Equation E.1:

$$P(A|L) = e^{-C[L,A]} \tag{E.3}$$

This frame based dtw algorithm is now extended in order to reflect the bigram probabilities which were used during the alignment process (see Equation 5.1). In other words, probabilities of substitution, deletion or insertion of phonemes depend on the context. Given that sequence xy is observed in an aligned file, the probability that this sequence corresponds to the labeled segment ab is denoted by $P(ab|xy)$. Similarly, if the sequence xy occurs in an aligned file, the probability that this sequence corresponds to the labeled

segments a is denoted by $P(a|xy)$. In this case the probability that label y is an insertion depends on the context of a preceding label x. Similarly, if the sequence x occurs in an aligned file, the probability that b is deleted when the preceding label a was recognized as x, is denoted by $P(ab|x)$. This results in the following definitions:

$$\begin{aligned} \mathbf{P(sub2)} &= P(ab|xy) \\ \mathbf{P(del)} &= P(ab|x) \\ \mathbf{P(ins)} &= P(a|xy) \end{aligned} \tag{E.4}$$

Specifically, these values are calculated by dynamic time warping the labeled with the corresponding aligned file using the frame-based method described above. As a result one can obtain the number of occurrences where segments xy correspond to ab and the number of occurrences of xy. Then, $P(ab|xy) = num(ab \wedge xy)/num(xy)$. P(del) and P(ins) are calculated in the same manner. In order to compensate for the sparseness of data, these bigram values are interpolated with unigrams by a factor λ.

$$\begin{aligned} \mathbf{P(sub2)} &= \lambda P(ab|xy) &&+(1-\lambda)P(a|x) * P(b|y) \\ \mathbf{P(del)} &= \lambda P(ab|x) &&+(1-\lambda)P(a|x) * P(b|x) \\ \mathbf{P(ins)} &= \lambda P(a|xy) &&+(1-\lambda)P(a|x) * P(a|y) \end{aligned} \tag{E.5}$$

The trust λ that is given to the bigram probabilities P is expressed by this measure which is related to the weighted entropy or information gained about the source P.

$$\lambda = \gamma + P \, log \, P \tag{E.6}$$

where P is any of $P(ab|xy)$, $P(ab|x)$ or $P(a|xy)$. γ can vary between 0.4 and 1.0 and denotes the maximum trust given to the bigrams. The corresponding graph of λ as a function of P is depicted in Figure E.4 Intuitively, one can imagine if P is one or zero, the training set is consistent within itself. The bigram information can be trusted and the value of λ is at its maximum. On the other hand if xy is aligned to ab only half the time, then there is an inconsistency in the data which may be due to the lack of data and is compensated for by emphasizing the unigram information.

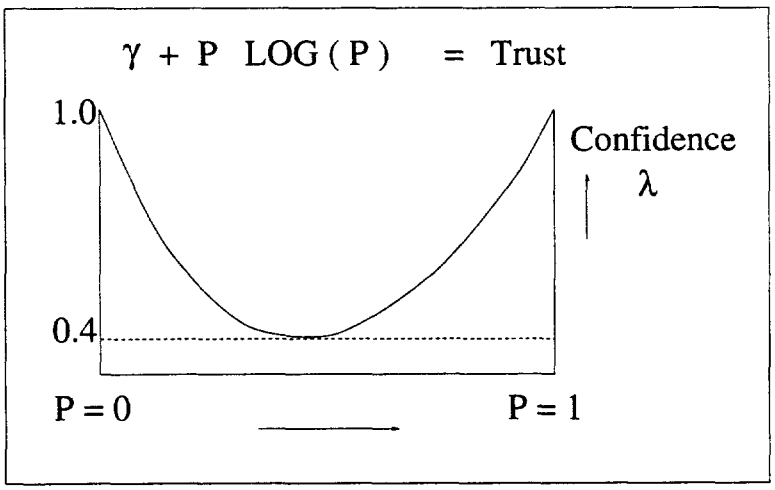

Figure E.4: Plot of confidence for bigram probability as a function of $P(L|A)$.

With these definitions, the dynamic time warping can now be defined with bigram probabilities to calculate the distance score β between two sequences **A** of length i and **L** of length j. Initializing as follows,

$$\begin{aligned} \beta[L_2, A_2] &= -log(P(sub2)) \\ \beta[L_1, A_2] &= -log(P(ins)) \\ \beta[L_2, A_1] &= -log(P(del)), \end{aligned} \tag{E.7}$$

the recursive definition is given as:

$$\beta[L_j, A_i] = max \begin{cases} \beta[L_{j-1}, A_{i-1}] & + P(sub2) \\ \beta[L_{j-1}, A_i] & + P(del) \\ \beta[L_j, A_{i-1}] & + P(ins) \end{cases} \tag{E.8}$$

The score leads to the probability that a given sequence **A** is a variation of the Center sequence **L**, as given in the next equation. Let **M** denote the Manhattan distance measure of the best paths between the labeled and aligned sequence; then the probability that aligned sequence **A** corresponds to the labeled sequence **L**, $P(L|A)$ is given as:

$$P(L|A) = \exp \frac{\beta[L_j, A_i]}{M} \tag{E.9}$$

Having derived $P(L|A)$, we now have a measure of closeness for any two sequences. The **Radius** is measured with this score. It determines which sequences may be associated with each other to form the different pronunciations of a single sequence given by the **Center**. This measure will also serve as part of the sorting parameter by which sequences are sorted with respect to their distance from the center sequence. This parameter is important because it determines the order in which sequences are associated with the **Center** and thereby influences the size of the **Radius**. This issue will be addressed in more detail below.

E.4 Weighting Factor

In order to overcome some of the shortcomings of the alignment process I will attempt to reestimate the probability that a given aligned sequence **A** corresponds to a labeled

sequence **L**, where both **A** and **L** occur in the aligned files. By adding up all sequences in the aligned files and multiplying their occurrence frequency by $P(L|A)$, one can approximate the actual occurrence of the labeled sequence. If N is the number of sequences associated with a center **L**, then $P(L)$ the estimated frequency occurrence of **L** as based on the frequency $P(i)$ of N sequences in the aligned files:

$$P(L) = \sum_{i}^{N} P(L|i)P(i) \tag{E.10}$$

The number of sequences N, that are associated with a given Center is determined by estimating the discrimination error due to their union. In order to estimate the language discrimination error due to a given set, the Bhattacharyya distance is used as described in Section 3.2. As with exact sequences, inexact sequences are represented by a mean and variance. The distribution of a set of sequences is calculated in order to estimate the error due to their union. Assuming normal distribution of occurrence frequencies of sequences, we can add up the sum of random variables to create $N(\mu, \sigma)$, where μ and σ here correspond to the mean and variance respectively of a "word" in terms of its associated sequences.

$$\begin{aligned} \mu_L &= \textstyle\sum_i^N P(L|i)u_i \\ \sigma_L &= \sqrt{\textstyle\sum_i^N P(L|i)^2 s_i^2} \end{aligned} \tag{E.11}$$

This formulation assumes independence of features which was shown in Chapter 4 not to be true. Therefore the mean and variance are also directly reestimated from the data, treating the list of sequences to be associated as one. The Bhattacharyya distance is then applied to these mean and variances derived from the data given by:

$$\frac{1}{2}e^{-\frac{1}{4}[\frac{(\mu_2-\mu_1)^2}{\sigma_1^2+\sigma_2^2}]} \tag{E.12}$$

This measure can be calculated at each successive level of associating sequences 1 through N with the **Center**. Figure E.5 shows how sequences, **A**, with high probability $P(L|A) = \beta$ are associated with the **Center**, **L**, and ordered with respect to β. With each newly associated sequence, according to the ordering, the new distribution parameters are calculated and the error is estimated. In Figure E.5, the minimal error, corresponding to maximal discrimination is achieved after adding **Sequence 3** to the **Center**. This therefore corresponds to the allowed **Radius**.

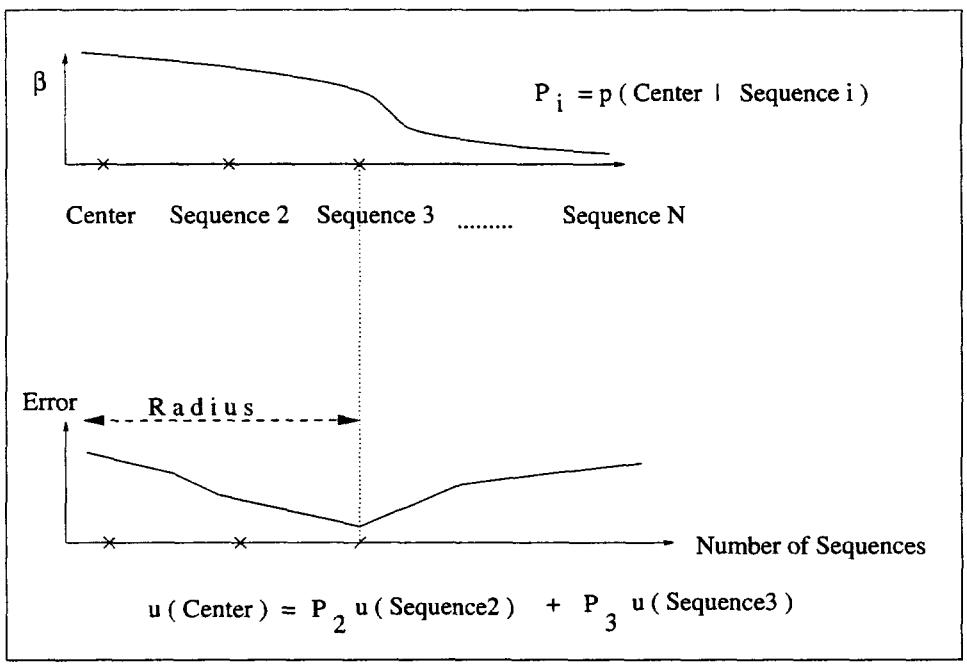

Figure E.5: Grouping sequences sorted by $P(L|A) = \beta$.

E.5 Feature selection

It was shown how sequences are related to each other by a distance score in Section E.3 and how they are each weighted to represent a word in Section E.4. The error function determines to what degree of inaccuracy sequences may be matched without sacrificing discriminability. The next step is to find the words, sort them by their usefulness and choose a set of features among them. The ideal word is one that has a **Radius** which allows a reasonable amount of matches in every incoming utterance. By allowing all sequences seen in the training set to be **Centers**, the system can evolve word representations automatically. For each such **Center** a list of N sequences is sorted by distance from the **Center**. For each sequence, including the **Center**, the error is estimated based on the combined set of sequences as seen in Figure E.5. The **Radius** is initialized to zero and is moved outward according to the ordered sequences. The final **Radius** is chosen at the point where a minimum discrimination error is estimated. This process of limiting the **Radius** reveals the importance of the order in which sequences are added to a set. This can be illustrated in a short example. Suppose the probability that the aligned sequence of labels /A/t/ corresponds to the labeled sequence /I/C/ is 50%. Suppose further the probability that the aligned sequence of labels /I/C/ corresponds to the labeled sequence /I/C/ is 10%. According to Formula E.10, the occurrence frequency $U(/I/C/)$ of the labeled sequence /I/C/ is estimated from the occurrence frequency of the aligned sequences u as:

$$U(/I/C/) = .50 * u(/A/t/) + .10 * u(/I/C/)$$

While this is true, sorting the sequences in this manner encourages using rare data thus causing bad agreement between training and test set and counteracting the goal of generalization. To clarify, assume that $u(/A/t/) = 2$ and $u(/I/C/) = 300$. The chances of

seeing the sequence $/A/t/$ in the test set are minimal compared to seeing $/I/C/$. In order to rectify this problem the sequences associated with the **Center**, **L**, are sorted according to $P(L|A)P(A)$ thus taking the occurrence frequency of **A** into account. That is the ordering is changed to sort by $P(L|A)P(A)$ while the weight of each sequence continues to be $P(L|A)$. Since $.50 * 2 = 1$ and $.10 * 300 = 30$, $/I/C/$ is a sequence which is much more important due to its frequency of occurrence. It is sorted closer to the **Center**, whereas $/A/t/$ is a rare sequence that will decrease in importance with respect to the **Center**. Now we have:

$$U(/I/C/) = (.10 * u(/I/C/)) + ... + (.50 * u(/A/t/)) + ...$$

The new order is much more robust because it encourages matching of sequences occurring frequently in the aligned files.

E.6 Language Identification

To identify the language of an utterance, we proceed as follows. Each incoming sequence is matched to all centers which are represented by a sequence and a radius. If the returned score of the match is within the given radius, then the corresponding word count is incremented. This method allows for more than one match at a time. All occurrence frequencies of these centers are normalized by the length of the utterance. Since the normal assumption that was used during error-estimation for clustering may not be appropriate, we use a non-linear neural network as classifier which is able to take co-occurrence of input features into account. For each of the sequences to be spotted the mean occurrence frequency of each selected sequence is used as a feature. λ was set to 1.0 after varying it showed no significant impact on sequence selection. We train a neural network in order to learn the discriminant function based on features derived after having seen 300 phonemes to build a representative statistic of the utterance.

E.7 Results

In Chapter 4 it is predicted that using inexact sequence matching should be no better than using exact sequence matching. This seems to be reflected in the results when using the method described in this chapter. When using the top 50 "words", the neural network is trained to discriminate between English and German with 11 hidden nodes. Evaluated on the same test set as used in Chapter 5 correct discrimination is 86%. The plot of language identification as a function of time is given in Figure E.6. After experimenting with a large number of choices for the various parameters and algorithms used for inexact matching, we now believe that this truly is a property of our system – as predicted by theory, inexact sequence matching does *not improve* over exact matching. The probability that the two systems using 59 phonemes for exact and inexact matching are the same is 38% using the two tailed test which means that they are not significantly different.

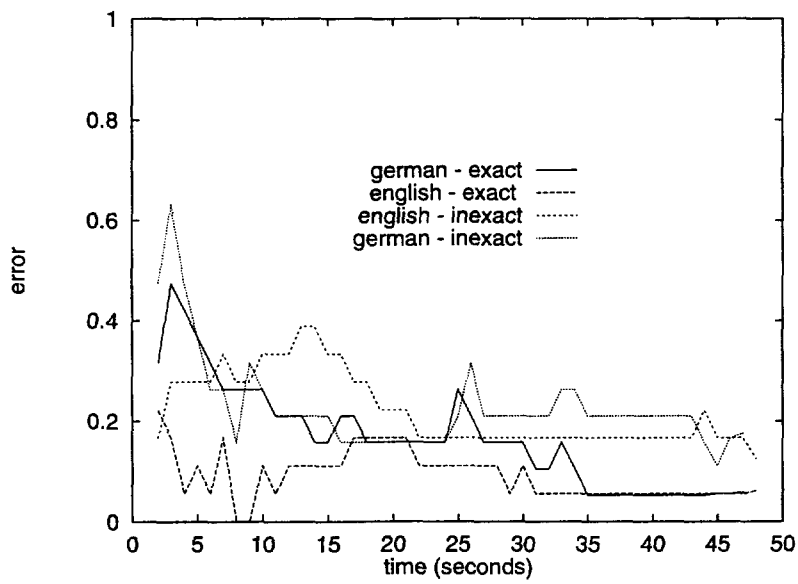

Figure E.6: Comparing results for exact vs. inexact string matching. Using 50 features results are plotted for the test set of German and English.

Biographical Note

Kay Berkling was born on October 20, 1967, in Mount Kisco, New York, USA. She grew up in Bonn, Germany, where she graduated from the Ursulinen Schule with a high-school degree in 1984. She then moved to Syracuse, New York where she graduated from Fayetteville-Manlius Highschool in 1986 and was admitted to Syracuse University. In 1991 she received her Bachelor of Arts & Science degree majors in German and French, her Bachelor of Science degree in Mathematics, and her Bachelor of Science degree in Computer Engineering with honors. She also earned a minor in Political Science while studying in France. Since 1991 she has been a research assistant at the Oregon Graduate Institute of Science & Technology as a member of the Center for Spoken Language Understanding. Her research interests include automatic language identification, accent and dialect adaptation, porting systems to other languages, and computer aided language teaching.

Publication List

1. Kay M. Berkling and Etienne Barnard, "Language Identification with Inaccurate Sequence Matching" *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, October, Philadalphia, USA, 1996, vol 3, pp. 1796-1799.

2. Kay M. Berkling and Etienne Barnard, "Theoretical Error Prediction for a Language Identification System using Optimal Phoneme Clustering" *Proceedings of the Fourth European Conference on Speech Communication and Technology*, September 18-21, Madrid, Spain, 1995, vol. 1 , pp. 351-354.

3. Muthusamy, Y.K., Kay M. Berkling, T. Arai, R. A. Cole and E. Barnard, " A Comparison of Approaches to Automatic Language Identification Using Telephone Speech," *Proceedings 3rd European Conference on Speech Communication and Technology (Eurospeech 93)*, Berlin,Germany, September, 1993, vol. 2, pp. 1307-1310.

4. Kay M. Berkling, Takayuki Arai, Etienne Barnard and Ronald .A. Cole, "Analysis of Phoneme-Based Features for Language Identification," *Proceedings of the 1994 International Conference on Acoustics, Speech and Signal Processing*, Adelaide, South Australia, April 19–22, 1994, vol.1, pp. 289-292.

5. Kay M. Berkling and Etienne Barnard, "Language Identification of Six Languages Based on a Common Set of Broad Phonemes", *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Yokohama, Japan, October, 1994, vol. 4, pp.1891-1895.

6. Yonghong Yan, Kay M. Berkling, and Etienne Barnard, "Bigram Models and Phoneme Clusters for Language Identification", *Proceedings of Speech Research Symposium (SRS)*, Johns Hopkins, Baltimore, USA, June, 1994, pp.22-30.