A STUDY ON THE USE OF CONDITIONAL RANDOM FIELDS FOR AUTOMATIC SPEECH RECOGNITION

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Jeremy J. Morris, B.S., M.A., M.S.

Graduate Program in Computer Science & Engineering

The Ohio State University

2010

Dissertation Committee:

Prof. Eric Fosler-Lussier , Adviser

Prof. Chris Brew

Prof. Mikhail Belkin

ABSTRACT

Current state of the art systems for Automatic Speech Recognition (ASR) use statistical modeling techniques such as Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs) to recognize spoken language. These techniques make use of statistics derived from the acoustic frequencies of the speech signal. In recent years, interest has been rising in the use of phonological features derived from these acoustic frequency features in addition to, or in place of, the acoustic frequency features themselves. These phonological features are derived from the manner that speech is physically produced in the vocal tract of the speaker, rather than models of how speech is heard by the listener.

Integrating phonological features into ASR models presents new challenges. The mathematical assumptions made to build current models may work well for features derived from acoustic frequencies, but do not necessarily fit phonological features as nicely. Explorations into how to alter the mathematical models to allow for this new type of input feature is an ongoing area of ASR research. This dissertation examines the use of the statistical model known as a Conditional Random Field (CRF) for ASR using phonological features. CRFs are statistical models of sequences that are similar to HMMs, but CRF models do not make any assumptions about the independence or interdependence of the data being modeled.

This dissertation provides (1) a CRF-based pilot system is able to achieve superior performance in a phonetic recognition task to a comparably configured HMM model, and

achieve this performance with many fewer parameters, (2) an extension of this model to create new features for an HMM-based system for word recognition, and (3) a fully developed system for word recognition using CRFs.

For Christine and Connor

ACKNOWLEDGMENTS

The work in this dissertation would not have been possible without the support of my adviser, Dr. Eric Fosler-Lussier. I am grateful to have had the opportunity to work in his lab under his guidance. His advice and insight have both helped to shape my professional growth and made my graduate work an enjoyable and interesting experience. In addition, I would also like to thank Dr. Chris Brew for his support and feedback during my graduate career and throughout the dissertation process. His questions often provided a unique perspective on problems that I might not have considered without his insights, and I am grateful to have had the opportunity to learn from him.

The members of the OSU Computer Science and Engineering AI group and the members of the OSU Clippers seminar were both important to the work in this dissertation. Their feedback, as well as their sharing of their own work, helped me to find new insights into my own work. I would especially like to express gratitude to the members of the OSU SLaTe lab, both past and present, who have assisted me in countless ways during this process – Tim Weale, Ilana Heinz, Rohit Prabhavalkar, Preethi Jyothi, Billy Hartmann, Josh King, Darla Shockley, Prateeti Mohapatra, Anton Rytting, Laura Stoia, and Tiangfang Xu. Whether listening to me ramble when getting my thoughts in order, providing a check on my logic, or just sharing complaints with each other over lunch, they helped me in countless ways over the years and their assistance is much appreciated.

Various portions of this work were funded under the auspices of the National Science Foundation and the Dayton Area Graduate Studies Institute. Their support for scientific research in general, and the support provided for this work in particular, is much appreciated. I would especially like to thank the NSF for providing the funding for the project that originally led me to work in the area of discriminative models for speech recognition – it is not too much of a stretch to say that without that funding this dissertation would likely be a very different one.

A number of friends have helped to keep me grounded over the years, and they all deserve thanks. I am very grateful to my good friend Tyler Heichel, who was willing to listen to me ramble on about nothing in particular over lunch and who I can't recall complaining once about it. I am also thankful to my irregular gaming group including Tyler, Andrew Lee, Paul Roethele, Ryan Green, Dave Mansbach and Chris Bernard who were often willing to help me blow off some steam with a game on various Sunday afternoons over the last few years. I would also like to thank Melanie Lehman, who despite not liking games, has been a wonderful friend and a big help through this entire experience to me and to my family.

Finally, but most importantly, I would like to thank my wife, best friend, and partner Christine for her support, understanding and assistance through this entire graduate school process. Without her constant encouragement and support over the years, none of this could have been accomplished. I am very glad to have found someone who can keep me grounded while still encouraging me to push myself as much as I have needed to over the years.

VITA

May, 1996 ..... B.S., Computer Science & Mathematics
Bowling Green State University
Bowling Green, OH, USA

June, 1998 ..... M.A., Education
The Ohio State University
Columbus, OH, USA

May, 2007 ..... M.S., Computer Science & Engineering
The Ohio State University
Columbus, OH, USA

PUBLICATIONS

Journal Articles

Jeremy Morris and Eric Fosler-Lussier. Conditional Random Fields for Integrating Local Discriminative Classifiers. *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

Conference Papers

Jeremy Morris and Eric Fosler-Lussier. Crandem: Conditional Random Fields for word recognition. *Interspeech*, 2009.

Eric Fosler-Lussier and Jeremy Morris. Crandem systems: Conditional Random Field Acoustic Models for Hidden Markov Models. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

Jeremy Morris and Eric Fosler-Lussier. Further experiments with detector-based Conditional Random Fields in phonetic recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

Jeremy Morris and Eric Fosler-Lussier. Combining phonetic attributes using Conditional Random Fields. *Interspeech*, 2006.

Jeremy Morris and Eric Fosler-Lussier. Discriminative phonetic recognition with Conditional Random Fields. *HLT-NAACL Workshop on Compuationally Hard Problems and Joint Inference*, 2006.

FIELDS OF STUDY

Major Field: Computer Science and Engineering

Studies in Automatic Speech Recognition: Prof. Eric Fosler-Lussier

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1: INTRODUCTION

One of the more common themes in the recent Automatic Speech Recognition (ASR) literature has been the re-envisioning of the appropriate input to statistical models. In particular, local posterior estimates, such as the prediction of phone classes given acoustic input, have been used to supplant or augment the traditional Mel-Frequency Cepstral Coefficient (MFCC) input [25]. Interest has also been shown in the idea of using sub-phonetic phonological (or articulatory) attributes[1] in ASR. It has been proposed (most notably in [49]) that the 'beads-on-a-string' approach to modeling speech as a connected sequence of distinct phone segments does not properly address pronunciation variability found in spontaneous speech. It has also been argued [35] that ASR systems can be improved by integrating statistical modeling techniques with more linguistically-directed feature extraction and recognition methods. These arguments point to an idea of modeling speech as connected sequences of interacting features rather than individual phone segments.

As acoustic representations based on linguistic knowledge are derived and extracted from the speech signal, methods must be examined to integrate these inputs to recognize speech. While models such as Hidden Markov Models and the more general Dynamic Bayesian Networks have been explored for this task, both models have a set of independence assumptions on the extracted features that either require an explicit decorrelation

[1]Traditionally, these have been called phonological features (or articulatory features) in the linguistics literature, but this creates a confound when considering acoustic features, such as MFCCs, or the CRF feature functions described in Section 2.3. In order to avoid confusion these are referred to as phonological attributes in this dissertation. However, the term *feature* is used to generally mean any acoustic representation that is input to a statistical system; thus, posterior estimates of phonological attributes may be features.

step before the features can be used (in the case of HMMs) or require the modeler to explicitly describe all dependencies among possibly hidden features in the model (in the case of DBNs). These can both lead to complications as the types of features being integrated change - the former because decorrelation for modeling purposes may remove or change important information in the underlying feature streams, the latter because the interactions of a new feature with previously defined features in the model may not be well known or easily discovered.

The family of statistical models known as Conditional Random Fields (CRFs) have properties that set them apart from DBNs and HMMs that may be advantageous for ASR. Unlike HMMs, CRFs are discriminative models and do not attempt to model how the input sequences are generated. CRFs therefore do not place any independence requirements among input sequences across time or across individual input values. Unlike DBNs, CRFs allow for an arbitrary structure of dependencies among features to exist without the need for the modeler to determine the underlying structure.[2] These properties of CRFs make them an attractive model for integrating together linguistically derived features for speech recognition.

But where the CRFs present a model with desirable properties, they also bring forward new challenges for building speech recognition systems. The discriminative nature of the CRF model means that in order to use them for recognition, the generative methods of current state-of-the-art statistical speech recognition must be modified to accommodate this new model. Different approaches to handling this challenge can be undertaken – from attempting to find ways to use the CRF in an HMM paradigm to deriving a model for

[2]Technically, CRFs determine the interdependencies in a combination of *feature functions* within an exponential model; the CRF does not relieve the modeler from the challenge of designing an appropriate set of feature functions, some of which might express particular dependencies between features.

speech recognition to accommodate these new models directly. Both of these approaches for using CRFs in ASR are explored in the following chapters.

This dissertation explores the potential of the CRF as a statistical model for speech recognition, specifically focused on the idea of CRF models as tools to integrate together a variety of linguistically-derived acoustic features. Chapter 2 of this dissertation reviews the statistical model for ASR, discusses prior work in the area of linguistic knowledge based feature extraction and integration, and reviews the CRF family of statistical models. To demonstrate the potential for this discriminative model in ASR, Chapter 3 describes a pilot system for phone recognition. This pilot system is shown to achieve results superior to a maximum-likelihood trained HMM system for the task of phone recognition. In Chapter 4, a system for word recognition that uses the results of a CRF phone recognition system as input is also derived and evaluated. While this combined HMM-CRF system is shown to have superior performance on the phone recognition task than a standard HMM system or the CRF system, this performance does not carry over into the task of word recognition. The results of this system are analyzed to determine why this improved performance does not carry over. Chapter 5 derives and evaluates a model for full, continuous automatic speech recognition using CRFs. This new direct decoding model is shown to perform in the word recognition task comparably to a maximum-likelihood trained HMM system over the same set of input features. Finally Chapter 6 concludes this dissertation with a summary and a discussion of possible extensions to this work.

CHAPTER 2: STATISTICAL MODELING AND THE USE OF PHONOLOGICAL ATTRIBUTES IN ASR

State-of-the-art ASR systems make use of *phonemes* as labels for individual subword units both during training and in recognition. Phonemes are abstract units that describe a particular segment of speech that can be distinguished by contrast within words [32]. In contrast, the term *phone* is used to describe the actual realization of the phoneme when spoken. ASR systems train their likelihood models based on associations between the input auditory frequency vectors taken from a segment of speech and the phoneme label associated with that segment.

Phoneme labels, however, are not the smallest unit of speech that could be modeled. Each phoneme label represents a bundle of phonological attributes that describe how that phoneme contrasts with other phonemes in the language. A variety of methods exist for determining what these phonological attributes are and how they should be assigned. As an example, for consonant phonemes, a possible system of assignment for these phonological attributes might include the *place* and the *manner* of articulation. For vowel phonemes, these might include the *height* of the tongue in the mouth, the *front-back* position of the tongue in the mouth and the *roundness* of the lips.

Incorporating these attributes into a statistical model for ASR is not a simple task, and various different methods have been examined in recent years. This chapter reviews the literature and provides a summary of different methods for ASR using these features. This dissertation examines the use of a discriminative statistical model – the Conditional

Random Field – as a method for incorporating these attributes into the statistical ASR framework.

The purpose of this chapter is three-fold. First a brief review of the statistical model of ASR is given in Section 2.1 to provide a baseline for the experiments that follow. Next, Section 2.2 provides a brief description of phonological attributes, a summary of the arguments for the use of phonological attributes in ASR, and descriptions of previous attempts to use statistical models to integrate phonological features into ASR systems. Finally Section 2.3 discusses the family of statistical models known as Conditional Random Fields (CRFs), including training and decoding paradigms for these models.

2.1 Review of the Statistical Model of ASR

In an HMM-based speech recognition system, the goal is to find the best sequence of words given the speech signal input to the system. As discussed in more detail by Huang et al in [26], an HMM model does this by finding the sequence of words $\hat{\mathbf{W}}$ that maximizes:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}) \tag{2.1}$$

where $\mathbf{X}$ is the set of speech signal inputs to the system, typically a vector of acoustic frequency coefficients extracted from the speech signal based on models of the human auditory system, such as Mel-Frequency Cepstral Coefficients (MFCCs) or coefficients derived via Perceptual Linear Prediction (PLPs). The number of coefficients used in these systems can vary, but it is common practice to use the first 12 frequency coefficients plus the energy coefficient, as well as the first and second order derivatives of these coefficients.

Via Bayes Rule, Equation 2.1 is transformed into:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}) = \arg\max_{\mathbf{W}} \frac{P(\mathbf{X}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{X})} \tag{2.2}$$

As $P(\mathbf{X})$ is the same for all for all possible word sequences across the common input $\mathbf{X}$, it may be safely ignored in the computation of the maximal word sequence that fits the data:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{X}|\mathbf{W})P(\mathbf{W}) \tag{2.3}$$

In general, state-of-the-art speech recognition systems do not attempt to directly calculate the probability $P(\mathbf{X}|\mathbf{W})$ for each word in their vocabulary. While for small vocabulary systems tracking models of the acoustic signal for each word may be possible, as vocabulary increases this method poses scaling difficulties. Additionally, training word-level models does not fully exploit the commonalities of speech among words that is found at the phonetic level. To account for these facts, the $P(\mathbf{X}|\mathbf{W})$ term of Equation (2.3) is rewritten as:

$$P(\mathbf{X}|\mathbf{W}) = \sum_{\mathbf{\Phi}} P(\mathbf{X}, \mathbf{\Phi}|\mathbf{W}) \tag{2.4}$$

where $\mathbf{\Phi}$ is a sequence of sub-word phonetic units. Equation (2.4) marginalizes the probability of the acoustics given the word sequence over all possible phonetic sequences. An assumption is then made that the acoustics ($\mathbf{X}$) are independent of the word sequence ($\mathbf{W}$) given the phone sequence ($\mathbf{\Phi}$):

$$P(\mathbf{X}|\mathbf{W}) = \sum_{\mathbf{\Phi}} P(\mathbf{X}|\mathbf{\Phi})P(\mathbf{\Phi}|\mathbf{W}) \tag{2.5}$$

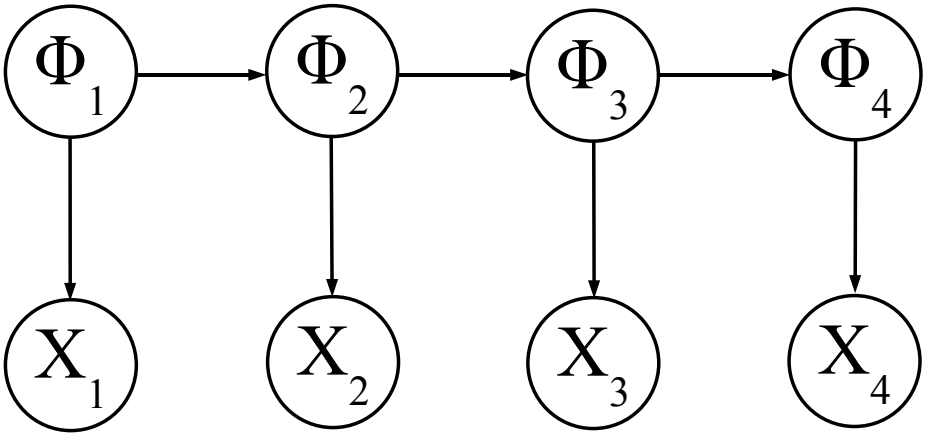

Figure 2.1: Graphical model of the Hidden Markov Model for ASR

In practice, Equation 2.5 is approximated with a *Viterbi approximation*, where the best phone sequence for each word sequence is used instead of marginalizing over all word sequences. This is substituted into Equation 2.3 to get the following equation:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \arg\max_{\mathbf{\Phi}} P(\mathbf{X}|\mathbf{\Phi})P(\mathbf{\Phi}|\mathbf{W})P(\mathbf{W}) \tag{2.6}$$

In this formulation, the likelihood $P(\mathbf{X}|\mathbf{\Phi})$ is called the *acoustic model*, the term $P(\mathbf{\Phi}|\mathbf{W})$ is the *dictionary model*, and the prior probability $P(\mathbf{W})$ represents the *language model*. The dictionary model is a relatively simple mapping of words to their phonetic sequences, and the language model is usually approximated with an n-gram language model.

The acoustic model in Equation 2.6 is typically implemented via a *Hidden Markov Model* (HMM) [55]. A graphical model of an HMM for acoustic modeling is shown in Figure 2.1[3]. Note that the HMM is described by two different probabilities. The first is

[3] Typically, HMMs for ASR are implemented using multiple states per phone to account for variation in the acoustics across time in the production of a phone. Multi-state models can be generalized from single state models, and are discussed more fully in Chapter 5, but for the purposes of discussion in this chapter single state models will be used as examples.

the *emission probability* – $P(X|\Phi)$ – the likelihood that a single frame of acoustics X was produced by the phone Φ. The second is the *transition probability* – $P(\Phi_t|\Phi_{t-1})$ – the probability of transitioning to the phone Φ_t given that the previous phone was Φ_{t-1}. Equation 2.7 shows how the acoustic model over the entire speech signal $\mathbf{X}$ can be decomposed into a product of the emission probabilities and the transition probabilities:

$$P(\mathbf{X}|\mathbf{\Phi}) = \prod_{t=1}^{T} P(X_t|\Phi_t)P(\Phi_t|\Phi_{t-1}) \tag{2.7}$$

Note that this decomposition of the likelihood requires an assumption of the independence of the feature inputs across time [55]. This assumption is also displayed in the graphical model provided by Figure 2.1 by the lack of connections between emitted feature vectors X. This assumption is not necessarily true in spoken language, where features across the speech signal the current realization may not be independent of the features in the previous (or successive) realization.

2.2 Phonological Attributes and Their Uses in ASR

When discussing sub-phonetic units, linguistic theory provides for different methods of breaking phones down into sub-phonetic units (known variously as "distinctive features", "phonological features", or "phonological attributes"). To provide some background for the discussion of the use of these linguistic features in ASR, a brief discussion of a few important examples of phonological attributes is provided in this section.

The system of distinctive features known as SPE was outlined by Chomsky and Halle in [6] (the term SPE refers to the title of their work – *The Sound Patterns of English*). Chomsky and Halle describe a phonetic representation of a segment of speech as a two-dimensional matrix where each row is a particular phonetic feature and each column is

one of the consecutive segments of the utterance (e.g. the realized phones). As originally formulated, each of the phonetic features defined in this system is *binary* in nature – the feature can take on either a positive value (e.g. '+', indicating the existence of the feature in the segment) or a negative value (e.g. '-', indicating the non-existence of the feature in the segment). For example, a feature indicating nasality is designated as [+nasal] for nasal segments (such as the phones /n/ or /m/) and [-nasal] for non-nasal segments. Each feature describes a single, binary attribute of some aspect of speech production – the position of the tongue in the mouth, the shape of the oral cavity, whether the vocal cords are vibrating, etc.

Note that in this formulation of SPE features, every feature must be characterized as either on or off for each phone segment. Later formulations of the SPE feature system allow features to be *univalent*, where a feature may only be characterized as 'on' for a given segment and no meaning is given to a feature being 'off'. This is the variant of SPE that is used in [21], where the place of articulation features LABIAL, CORONAL and DORSAL are defined as univalent features. In this system, binary features are allowed to be keyed off of particular univalent features and only allowed to take positive or negative values if the univalent feature they are associated with exists. For example, the univalent feature CORONAL in this system allows the use of the binary features anterior, distributed and strident. Note that for these phone labels, the univalent features DORSAL and LABIAL are only used to describe the phones that have the features – there is no such thing as a [-LABIAL] or [-CORONAL] feature, for example.

Various attempts to use phonological features in ASR have examined the use of multi-valued phonological feature classes, rather than the binary classes of SPE ([29], [30]). In a multi-valued feature framework, each set of features is grouped together into a distinct class

of features. Each class of features groups together features such that phones can be defined as a vector of feature values with one feature for each class. For example, the multi-valued system examined in [29] breaks features up into classes of Centrality, Continuant, Front-Back, Manner, Phonation, Place, Roundness and Tenseness. Each of these classes has between two and ten different features within it, and each phone can be described as a vector of eight features. The multi-valued system used in [30] has five classes: Voicing, Manner, Place, Front-Back and Rounding, and each phone can be described in a vector of five features. The multi-valued system used in the proposal has eight classes: Sonority, Voicing, Manner, Place, Vowel Height, Vowel Frontness, Vowel Roundness, and Vowel Tenseness. These systems are not necessarily derived directly from a particular phonological system, but are intended to cover the phonetic feature space of human speech in the manner of the IPA phonetics chart.

A multi-valued system directly models the interdependencies of features in a way that SPE-style features do not. In an SPE-style system using only binary features, each place of articulation is modeled as a set of binary features in different positive and negative combinations. The dental consonants (such as /t/ or /d/) would have the features [+anterior] and [+coronal] (among others) to define them. The SPE-style system used in [21] with univalent features would instead have the univalent [CORONAL] and binary [+anterior] features defined for dentals (among others). In contrast, a feature class for the Place features in a multi-valued system might only have a single feature [dental] defined for the dental consonants. The experiments in this dissertation are implemented using multi-valued systems of phonological features, but the ideas and models expressed in this work are not dependent on these kinds of feature systems. The models presented here can be fairly easily extended to SPE or other phonological attribute systems.

2.2.1 The Use of Phonological Attributes in ASR

In [30], Kirchhoff identifies four arguments for incorporating phonological attributes into ASR: more robust statistical models, the ability to better model co-articulation in speech, the ability to perform selective processing of the features, and noise robustness advantages. In [31], Kirchhoff provides another argument in favor of using phonological attributes is proposed by the same author: the dual nature of phonological features as both acoustic and linguistic units.

The argument for statistical robustness is a simple one – since phonological attributes are shared across multiple phone instances in a training corpus, phonological attribute classifiers have more data to train on than a phone classifier training on the same corpus. In addition, phonological attribute classifiers have fewer distinct classes of features to distinguish than phone classifiers. More training data per class and fewer overall classes therefore both lead to an overall better model. Experiments performed by Kirchhoff in [30] bear out this observation – in general, phonological attribute classifiers tend to have a higher accuracy than comparable phone classifiers. Counter to this argument, however, are experiments performed by Rajamanohar and Fosler-Lussier in [57] which showed that phonological attribute classifiers built by combining the results of phone label classifiers achieved a higher accuracy than classifiers built to directly classify phonological attributes.

The co-articulation argument is an argument from linguistic principles. Because phoneme labels are an abstract model for speech, a single label does not completely describe the variation that appears in a speech signal. Since speech is a continuous process, these phoneme labels do not exist independently of one another. Instead, the features that exist in the realization of a particular phone can be highly influenced by co-articulation of features from previous phones and from succeeding phones. For example, the vowel leading into the

nasal phone /m/ can acquire nasalized characteristics, due to the motion of the lips from open (to articulate the vowel phone) to closed (to articulate the labial nasal phone /m/). Pronunciation variation due to co-articulation can be expressed using context-dependent rules that describe changes to the features for a particular phoneme based on surrounding phonemes.

The selective processing and noise robustness arguments both come from the acoustic nature of the speech signal. The argument is that different phonological properties of the speech signal deteriorate to varying degrees under different environmental conditions. Kirchhoff uses the examples of voicing attributes, which are fairly robust to noise, and place features, which she claims deteriorate to a greater degree in the presence of noise. In an acoustic frequency framework, these differences are all conflated into the overall frequency coefficients and are all treated the same. In a phonological attribute framework, on the other hand, these differences can more easily be targeted and accounted for separately. Features that are less robust can be given more context, a different type of acoustic frequency extraction, or other adaptations to increase their robustness, while features that are already robust can be modeled more simply.

Finally, the argument for the "dual nature" of the phonological features is both an acoustic and a linguistic one. Kirchhoff argues that because phonological attributes have both acoustic correlates in the speech signal and a strong relationship to higher-level linguistic units, they provide a more fundamental link between acoustics and the lexicon of spoken language than other representations such as phonemes.

In [68], Stüker et al. present another use for phonological attributes – as part of a multilingual ASR system. Not all phonemes exist in all languages, and this fact prevents traditional ASR acoustic models trained in one language from being used in a recognizer

for another language. In contrast, phonological attributes, as a more fundamental unit of speech, are mostly shared across languages. The ability to share models across languages makes it easier to quickly produce new ASR systems for new languages, and makes phonological attribute models an attractive prospect for multi-language applications.

Different methods for extracting phonological attribute information from an acoustic speech signal have been explored in recent years. The use of multi-layer perceptron ANNs is popular in the literature ([30], [14], [5]). In this approach, neural networks are trained on the input acoustic frequency signals to classify these inputs according to the existence or non-existence of particular attribute. Similar approaches have been performed with Gaussian Mixture Models ([69],[40],[68]), support vector machines ([28]), dynamic Bayesian networks ([15]), and recurrent neural networks ([29]). The experiments discussed in this dissertation are built on a foundation of work that uses MLP ANNs to derive multi-valued attributes ([57]), and these attributes will be used in discussion. However, the overall goal of this work is to remain as neutral as possible on how attributes are derived and focus instead on how they may be combined for recognition purposes.

There have also been a number of different avenues explored in recent years for combining phonological attributes together for ASR. In [30], Kirchhoff uses the outputs of MLP ANNs are used as emission probabilities for a Hybrid HMM/ANN ASR system [42]. In this Hybrid HMM/ANN system, an ANN is used to combine together phonological features to determine phone label emission probabilities for the HMM. Tandem HMM methods [25], where neural network outputs are used as inputs to a Gaussian-based HMM, have also been examined as a method for ASR using phonological attributes. In addition to the Hybrid system, Kirchhoff also describes a system that uses outputs of a set of phonological attribute classifiers in a Tandem HMM system [30]; this style of system has been further explored

by Launay et al in [34] and by Cetin et al in [50].[4] As the Tandem HMM system is built using mixtures of Gaussians to describe state emission probabilities, either the correlated phonological attribute inputs must first be decorrelated before being fed into the system or the system must make use of full or semi-tied covariance matrices and suffer an explosion in parameters and required training data.

A multi-stream HMM architecture for integrating GMM phonological attributes with acoustic features for ASR is proposed by Metze and Waibel in [40]. In this multi-stream model, each attribute is represented by a separate stream. Equation 2.8 shows the form of a multi-stream acoustic model [73].

$$P(\mathbf{X}_t|\mathbf{\Phi}_t) = \prod_{s=1}^{S} P(X_{s,t}|\Phi_t)^{\mu_s} \tag{2.8}$$

In the multi-stream model of Metze and Waibel, each stream s contains the feature information for a single attribute, and likelihood scores for each attribute are computed separately. In addition, the traditional acoustic frequency features can be modeled as an additional stream separate from the phonological attributes. The likelihood scores for each of these streams are weighted according to a stream weight μ_s and are then multiplied together to obtain the final likelihood of the phone label given all of the stream information. In this dissertation a multi-stream framework to integrate features is not necessary – the CRF framework allows multiple feature sets to be concatenated together and input directly as a single stream. In addition, this study is performed solely with the phonological and phone class feature outputs and does not integrate acoustic features (though work done by

[4] It is common, but not required, for Tandem systems to operate on inputs that include the acoustic features appended to the outputs of the MLP classifiers. In this paper, when we use a Tandem system we are describing a system that only uses the MLP classifier outputs and does not directly make use of the acoustic features.

Gunawardana et al in [20] shows that CRFs can also be built using acoustic features as inputs).

As an alternative to HMMs, Dynamic Bayesian Network models (DBNs) are used by Livescu et al in [38] to combine together phonological attributes for recognition. DBN models allow the structures of the dependencies of features to be explicitly modeled for training and recognition, but require that the modeler either know the structure underlying dependencies of the features, or learning these dependencies from training data. DBNs also require a more complicated Bayesian inference procedure for decoding than HMMs, due to the extra dependencies in the model.

One can also choose to model phonological attributes directly within HMMs by effectively expanding the state space of the HMMs to represent combinations of phonological attributes. In [8, 70], Deng and Sun use overlapping phonological attribute bundles as states in HMMs, forgoing the traditional triphone model by explicitly incorporating prosodically sensitive rules describing how phonological attributes interact. Such an approach, while requiring extensive development in encoding phonological rules, achieves a good result (72.95% on the full TIMIT phone classification test set compared to 70.86% using a standard triphone system).

2.3 Conditional Random Fields

Conditional Random Fields (CRFs) were introduced as a discriminative model for modeling data structured as Markov random fields by Lafferty et al in [33]. Although CRFs can be created that handle an arbitrary graphical structure, this dissertation restricts itself to considering a particular class of CRFs known as *Linear-Chain CRFs*. Linear Chain CRFs are a subset of the CRF family of models assume that the data can be modeled as a sequence

of labels with a Markov assumption that each label is dependent only on the label immediately previous and immediately following and the observations given to the model. When this chain structure is applied to speech, the nodes can be considered to be labels across a time sequence, dependent only on the phone labels immediately prior and immediately following the current label, as well as the input speech signal.

The uses of linear chain CRFs have been previously explored in tasks such as part of speech tagging [33] and parsing [62]. In the ASR domain, CRFs have shown impressive results in the area of phone classification, as described by Gunawardana et al in [20] and Yu et al in [74], and phone recognition as described by Abdel-Haleem in [1]. These works share some similarities with the work in this dissertation in that all of these works are concerned with the use of the application of CRF models for evaluating acoustic information. There are some key differences, however. Of note is that the work performed by Gunawardana et al and Yu et al focuses on the use of CRFs to directly model phone probabilities directly over extracted acoustic frequency features, while this work examines CRFs as a model for using extracted linguistic features of the acoustic frequency features for recognition purposes. Another key difference is that the work performed by Gunawardana et al and Yu et al explores the use of hidden state sequences for modeling the phones being classified (e.g. Hidden Conditional Random Fields or HCRFs), while the work here uses labeled phones with no hidden state sequences for training. Finally, work performed by Gunawardana et al and Yu et al both focus only on the task of phone classification, while the work in this dissertation initially examines the task of phone recognition and expands on these experiments to full word recognition. In the phone classification task the CRF is given phone boundaries and asked only to identify the phone that exists between the two bounds. The phone recognition task is a slightly harder task that asks the CRF to postulate an entire

phone sequence given an input speech signal, and so is not given boundary information for the phones involved.

The work performed by Abdel-Haleem in [1] is closer in nature to the phone recognition work described in Chapter 3, though there are differences. Abdel-Haleem also evaluates on the task of phone recognition, but the input space for the CRF models used in his work is a sparse vector of input features derived from Gaussian likelihood scores for individual Gaussians from the Gaussian mixture models generated for individual phones, while this work uses dense vectors of input values derived from MLP neural network outputs for both phones and phonological features. Additionally, the work performed in this dissertation extends on the phone recognition models to provide a CRF-based model for word recognition.

More recent work performed by Zweig and Nguyen in [76] makes use of segmental CRFs for continuous speech recognition. Unlike the work in this dissertation, the work by Zweig and Ngyuen does not attempt to use the CRF directly over the frame-level acoustic information. Instead Zweig and Nguyen use an approach that takes the output of an HMM-based ASR system for use as input features along with n-gram language model features and other pronunciation features to perform word-level recognition using a segmental CRF. Their system is shown to provide an improved performance for voice search query word recognition over the baseline MLE-trained HMM system that provides features to the CRF. This dissertation examines the use of CRF models at the acoustic level, and proposes a method for word recognition using these CRF acoustic models that is more in line with traditional statistical ASR techniques than the Zweig and Nguyen framework. However, the framework derived by Zweig and Nguyen does not rely on an HMM system for its

input features, but rather for a system that provides features appropriate for the segmental CRF.

As stated above CRFs are *discriminative* models, but it should be understood that there are different ways that the term *discriminative* is used in the ASR literature and at what level the CRF should be considered a discriminative model. The most obvious use of the word *discriminative* in this context lies in its membership in the family of *discriminative statistical models*, in contrast to the family of *generative statistical models* which contain models such as HMMs. Where a generative model uses the likelihood of the data and a model prior to compute class posteriors, a discriminative model attempts to compute the posterior of observed data directly, without modeling the way that the data has been generated explicitly. A specific accounting of this generative/discriminative dichotomy is given in detail by Sutton and McCallum in [18].

Another way that the term *discriminative* is used in ASR literature is in the context of *discriminative training methods*. While generative models, such as HMMs, can be trained using a non-discriminative criterion (Maximum Likelihood), they can also be trained via a number of discriminative criteria such as Maximum Mutual Information (MMI) or Minimum Phone Error (MPE)[48, 27, 63, 64, 53, 52, 51]. In this case the term *discriminative* refers to the criterion used for training is attempting to maximize the discrimination between competing classes, even though the underlying model is a generative statistical model. Discriminative statistical models have this training criterion inherent in the model itself – any training criterion for a discriminative model will attempt to maximize the discrimination between competing classes. In this work the CRFs are trained using a Conditional Maximum Likelihood (CML) training criterion, though the use of others (such as MPE) can be imagined.

Finally, the term *discriminative* can also be applied to the *features* used to train the statistical models, such as the discriminative phone and phonological attribute classifier output discussed above. These types of features are independent of the overarching statistical model used for integrating them for ASR – non-discriminative acoustic model features have been used in discriminative CRF models (e.g. HCRFs [20]), while discriminative phone classifier outputs have been used in generative HMMs (e.g. Tandem HMMs [25]). In addition, it is also quite common to concatenate non-discriminative acoustic features with discriminative features in a generative Tandem HMM (as in [50]).

2.3.1 Model

A Conditional Random Field (CRF) is a probabilistic model that directly models the posterior distribution of a label sequence conditioned on the observed data presented to it. Unlike a Hidden Markov Model, which attempts to model how the observed data is generated to select the most appropriate label, a CRF is a discriminative model that instead uses attributes of the observed data to constrain the probabilities of the various labels that the observed data can receive.

A CRF defines a posterior probability $P(\mathbf{y}|\mathbf{x})$ of a label sequence $\mathbf{y}$ for a given input sequence $\mathbf{x}$. In a linear chain conditional random field, the label for a given frame depends jointly on the label of the previous frame, the label of the succeeding frame, and the observed data $\mathbf{x}$. These dependencies are computed in terms of functions defined by pairs of labels and by label-observation pairs. The input sequence $\mathbf{x}$ corresponds to a series of frames of speech data, while the label sequence $\mathbf{y}$ is the series of labels assigned to that observed frame sequence. Each frame in $\mathbf{x}$ is assigned exactly one label in $\mathbf{y}$.

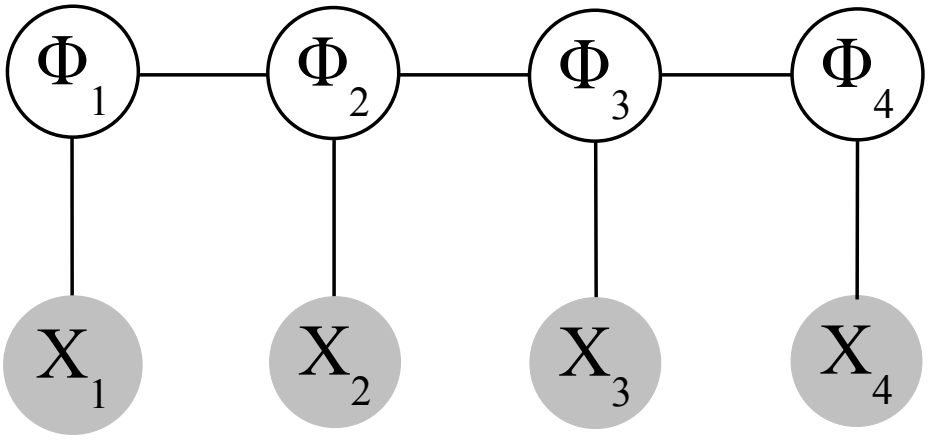

Figure 2.2: Graph of a Linear Chain Conditional Random Field

An example linear chain conditional random field graph is shown in Figure 2.2. Here each phonetic label for a particular time segment is specified by a node labeled with Φ and each observation for a particular time segment is represented as a node labeled with X. Note that the CRF formulation does not assume any particular relationship among the observed data nodes – the nodes of observed data may be connected in any arbitrary manner and the same formulation may be used. What follows is a short summary of the CRF model and its derivation as presented originally in [33] for discussion purposes.

Lafferty *et al.* [33] define a CRF in terms of its graph structure, which describes the Markovian structure of the independence assumptions in this undirected probabilistic model. Unconnected nodes in the graph are independent given the intervening nodes. When the graph is a linear chain of nodes (such as those representing labels on individual frames of speech, as in Figure 2.2), the cliques of the graph (edges and vertices) can be used to define a probability distribution by the Hammersley-Clifford theorem of Markov

random fields [4]. In the linear-chain graph, the distribution of the label sequence $\mathbf{y}$ given the observation sequence $\mathbf{x}$ will have the form:

$$P(\mathbf{y}|\mathbf{x}) = \frac{\exp \sum_t (\sum_i \lambda_i f_i(\mathbf{y}, \mathbf{x}, t))}{Z(\mathbf{x})} \tag{2.9}$$

where t ranges over the frame indices of the observed data and $Z(\mathbf{x})$ is a normalizing constant over all possible label sequences of $\mathbf{y}$ computed as:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} (\exp \sum_t (\sum_i \lambda_i \cdot f_i(\mathbf{y}, \mathbf{x}, t))) \tag{2.10}$$

The CRF is thus described by a set of *feature functions* (f_i), defined on graph cliques, with associated weights (λ_i). A feature function is non-zero only if the labels associated with the function match the labels in the sequence $\mathbf{y}$ for the observation at time t and the observation in $\mathbf{x}$ at time t shows the evidence required for the feature function. In a linear-chain CRF, two different broad types of feature functions are usually defined: *state feature functions*, associated with the graph vertices, whose output is dependent only on the observations and the label at the current timestep t (corresponding to the nodes in the graph) and *transition feature functions*, associated with the graph edges, whose output is dependent on the observations and both the label at the current timestep t and the label at the previous timestep $t-1$ (corresponding to the edges of the graph). Breaking the functions f above up into these separate categories of *state* and *transition* feature functions, Equation 2.9 above can be re-written as:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{\exp \sum_t (\sum_i \lambda_i s_i(y_t, \mathbf{x}, t) + \sum_j \mu_j f_j(y_{t-1}, y_t, \mathbf{x}, t)}{Z(\mathbf{x})} \tag{2.11}$$

where s are *state* feature functions with associated weights λ and f are *transition* feature functions with associated weights μ. As stated above, a *state* feature function associates

the label of a single node at time t (denoted as y_t) with the set of observations $\mathbf{x}$. As an example *state* feature function, consider Equation 2.12 below (this model is described only abstractly here as an example of a feature function and will be returned to and more fully examined in the context of a CRF system for phone recognition in Chapter 3):

$$f_{/b/,voi}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} 1, & \text{if } y_t = /b/ \text{ and} \\ & voiced(x_t) = true \\ 0, & \text{otherwise} \end{cases} \tag{2.12}$$

This *state* feature function describes a feature where the CRF is considering the phone label $/b/$ for the node y_t. From the observation sequence $\mathbf{x}$, it considers whether there is evidence for voicing in the observation sequence at time t via the function $voiced(x_t)$. If the proposed label supplied to this function is $/b/$ and the voicing evidence $voiced(x_t)$ both hold, then this function returns a non-zero value. Otherwise, the value of this function is zero and it provides no positive support for the hypothesis that the label at y_t should be $/b/$. Similar functions could be crafted for every phone label in the inventory – positive correlations between observation and label (such as in voiced phones) will be represented by positive λ weights, negative correlations (as in unvoiced phones) can be represented by negative lambda weights and observation-phone pairs that are uncorrelated will have near-zero λ-weights (and for efficiency can be ignored).

Transition feature functions operate in a similar fashion, except that instead of attempting to characterize a link between an observation and a single node, *transition* feature functions characterize a link between an observation and a transition between nodes. As an example, Equation 2.12 can be extended to a transition feature function in the following manner:

$$f_{/b/,/ah/,voi}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} 1, & \text{if } y_{t-1} = /b/ \text{ and} \\ & y_t = /ah/ \\ & voiced(x_t) = true \\ 0, & \text{otherwise} \end{cases} \tag{2.13}$$

Here the *transition* feature function will have a non-zero value only in the case where a transition from the phone $/b/$ to the phone $/ah/$ is being hypothesized and there is evidence of voicing at time t in the observation sequence. Again, *transition* feature functions such as this one can be crafted for each pair of labels in the inventory, and associated weights provide for how important the observed evidence is for the existence of the label in the sequence.

$$f_{/b/,/ah/,bias}(\mathbf{y}, \mathbf{X}, t) = \begin{cases} 1, & \text{if } y_{t-1} = /b/ \\ & y_t = /ah/ \\ 0, & \text{otherwise} \end{cases} \tag{2.14}$$

A CRF model can be built where the *transition* feature functions are not supported by observations at all, but are only be implemented as a *bias* feature function. An example of such a function is given in Equation 2.14. Here the value of the function depends only on the values assigned to the labels in the current and previous time segments, rather than on the labels and evidence from the observation sequence. Bias functions such as these can also be implemented as *state* feature functions.

2.3.2 Training

CRFs are trained through maximization of the conditional likelihood function $P(\mathbf{y}|\mathbf{x})$ over a set of training data. Different approaches to training models of this type have been examined (see for example [62] and [39]). In [20], both quasi-Newton gradient descent and stochastic gradient descent (SGD) methods are shown to perform well for CRF training for phone classification. In this work, two forms of training are used: gradient descent via the

quasi-Newton Limited-Broyden-Fletcher-Goldfarb-Shanno (L-BGFS) algorithm following work performed in [62], as well as stochastic gradient descent, following the work performed in [20]. A comparison of these two methods is discussed in Chapter 3.

To use any gradient descent method, the gradient of the likelihood function must be calculated. For the purposes of discussion, as well as for use in Chapter 4, the derivation of the gradient as given in [62] is given here.

First, the feature functions are ordered into a vector of feature functions $\mathbf{f}$. Next, the *global feature vector* $\mathbf{F}$ of the input sequence $\mathbf{x}$ and the corresponding label sequence $\mathbf{y}$ over the entire sequence is computed as:

$$\mathbf{F}(\mathbf{y}, \mathbf{x}) = \sum_{t=0}^{T} \mathbf{f}(\mathbf{y}, \mathbf{x}, t) \tag{2.15}$$

This allows Equation (2.15) to be rewritten as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{\exp \lambda \cdot \mathbf{F}(\mathbf{y}, \mathbf{x})}{Z(\mathbf{x})} \tag{2.16}$$

where λ is the vector of weights corresponding to the feature function vector $\mathbf{f}$. The normalization value $Z(\mathbf{x})$ can be rewritten as:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \lambda \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}) \tag{2.17}$$

The log likelihood of a label-observation pair $(\mathbf{y}_j, \mathbf{x}_j)$ given the weight vector λ is then formulated as:

$$\mathcal{L} = \log \lambda \cdot \mathbf{F}(\mathbf{y}_j, \mathbf{x}_j) - \log Z(\mathbf{x}_j) \tag{2.18}$$

Taking the gradient of Equation (2.18) with respect to the weights λ yields:

$$\nabla \mathcal{L} = \mathbf{F}(\mathbf{y}_j, \mathbf{x}_j) - \sum_{\mathbf{y}} \frac{\mathbf{F}(\mathbf{y}, \mathbf{x}_j) \cdot \exp \lambda \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}_j)}{Z(\mathbf{x}_j)} \tag{2.19}$$

or equivalently

$$\nabla \mathcal{L} = \mathbf{F}(\mathbf{y}_j, \mathbf{x}_j) - \sum_{\mathbf{y}} \mathbf{F}(\mathbf{y}, \mathbf{x}_j) \cdot E_{P_\lambda(\mathbf{y}|\mathbf{x}_j)} \tag{2.20}$$

where

$$E_{P_\lambda(\mathbf{y}|\mathbf{x}_j)} = \frac{\exp \lambda \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}_j)}{Z(\mathbf{x}_j)} \tag{2.21}$$

is the probability of the sequence y given x_j and the likelihood of an entire training set of K label/observation pairs can be formulated as:

$$\nabla \mathcal{L} = \sum_{k=0}^{K} [\mathbf{F}(\mathbf{y}_k, \mathbf{x}_k) - \sum_{Y} \mathbf{F}(\mathbf{Y}, \mathbf{x}_k) \cdot E_{P_\lambda(\mathbf{Y}|\mathbf{x}_k)}] \tag{2.22}$$

It is obvious that to compute this gradient the term $E_{P_\lambda(\mathbf{Y}|\mathbf{x}_k)}$ must be able to be efficiently computed, as must the normalizing term $Z(\mathbf{x}_j)$. Fortunately, a variant of the forward-backward algorithm is derived in [62] which can compute both of these terms efficiently for linear-chain CRFs:

For a given sample $\mathbf{X}_k$, we seek to compute:

$$\sum_{y} \mathbf{F}(\mathbf{y}, \mathbf{X}_k) \cdot E_{P_\lambda(\mathbf{y}|\mathbf{X}_k)} \tag{2.23}$$

For each time step t we define a *transition matrix* $M_t[y, y']$ as:

$$M_t[y, y'] = \exp \lambda \cdot \mathbf{f}(\mathbf{y}, \mathbf{X}, t) \tag{2.24}$$

where $\mathbf{y}_{t-1} = y$ and $\mathbf{y}_t = y'$. In other words, every cell of the transition matrix M_t contains the state and transition features computed by moving from label y at time $t-1$ to label y' at time t.

Next, for each state or transition feature function at time t, we create the feature function matrix f as:

$$f_t[y, y'] = f(\mathbf{y}, \mathbf{X}, t) \tag{2.25}$$

where again $\mathbf{y}_{t-1} = y$ and $\mathbf{y}_t = y'$. We can rewrite the expression in (2.15) as follows:

$$\sum_y \mathbf{F}(\mathbf{y}, \mathbf{X}_k) \cdot E_{P_\lambda(\mathbf{y}|\mathbf{X}_k)} = \sum_t \frac{\alpha_{t-1}(f_t * M_t)\beta_i^T}{Z(X_k)} \tag{2.26}$$

Where:

$$\alpha_t = \begin{cases} \alpha_{t-1}M_t, & 0 < t \leq T \\ \mathbf{1}, & \text{t=0} \end{cases} \tag{2.27}$$

$$\beta_t = \begin{cases} M_{t+1}\beta_{t+1}^T, & 1 \leq t < T \\ \mathbf{1}, & \text{t=T} \end{cases} \tag{2.28}$$

$$Z(\mathbf{X}_t) = \alpha_T \cdot \mathbf{1}^T \tag{2.29}$$

Using this formulation, the gradient can be computed by taking a forward pass across the sequence of length T to accumulate the α forward values, and then taking a backward pass across the sequence to accumulated the β values. The gradient can then be computed on a per-sample basis using equation (2.19).

Training methods: Limited-BFGS and Stochastic Gradient Descent

Limited-BFGS (or L-BFGS) is a quasi-Newton method for gradient descent that has been shown to function well for training CRF and other exponential models in various domains ([62],[39]). L-BFGS is a batch method that first computes the gradient of the entire training set with respect to the current weights, then moves in small steps along the computed gradient to find a minimum for the gradient.

The stochastic gradient descent (SGD) method is described by Gunawardana et al in [20] as a method for training CRF models, and was found in that work to perform better than the L-BFGS method on speech data for phone classification. Unlike the L-BFGS method, the SGD method is an online training method that updates the λ-weight values after each presentation of a training sample. The form of the SGD λ-weight update is:

$$\lambda^{(n+1)} = \lambda^n + \eta^n U^n \log \nabla \mathcal{L}^n \tag{2.30}$$

where n is the n-th training sample presentation, η is the learning rate, and U is a conditioning matrix. Note that this formulation of the SGD formula is similar to the familiar perceptron learning rule, and training can be implemented in a similar manner. The conditioning matrix U is a square matrix, and this work follows Gunawardana et al [20] in assigning U to be the identity matrix and using a static learning rate η^n across all samples. There is no requirement for U to be an identity matrix [66], however using a non-identity conditioning matrix requires prior knowledge of how the various feature functions will interact with one another – off-diagonal elements will create dependencies among the various feature functions during the computation of the weight update. This work chooses to remain neutral in this regard and uses the identity matrix, but leaves open the possibility that

a better convergence could be acquired if a more complex conditioning matrix were to be constructed.

Additionally, the observation was made in [20] that this SGD technique attains better performance when the λ-weights given by Equation (2.30) are averaged across each presentation, rather than just using the final computed λ-weight. This technique has been shown in other areas to give an improvement in performance (e.g. [59],[7]):

$$\lambda_{avg} = \frac{1}{N} \sum_{n=1}^{N} \lambda^n \tag{2.31}$$

where n ranges over all of the training sample presentations (and hence over all of the λ-weight updates made during training).

2.3.3 Decoding

The decoding step involves finding the label sequence $\mathbf{q}$ over the data $\mathbf{X}$ that maximizes equation (2.15). Since the normalizing term $Z(\mathbf{X})$ is independent of the label sequence q this is equivalent to:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \lambda \cdot \mathbf{F}(\mathbf{y}, \mathbf{X}) \tag{2.32}$$

which can be found by decomposing $\lambda \cdot \mathbf{F}$ into a sum of the individual $\lambda * f$ values across time for all observations. The best path across time can then be found by application of the Viterbi algorithm.

2.4 Summary

This chapter has provided the foundation for the experiments examined in this dissertation. A review of the statistical model for ASR was provided in Section 2.1, and this

model will be re-examined and modified in successive chapters. An overview of linguistically motivated phonological attributes, their interest to ASR, and previous methods of using them in ASR systems was discussed in Section 2.2, providing a motivation for this work as a whole. Section 2.3 reviewed the CRF model – a sequential statistical model that this work examines for the task of ASR using diverse, highly correlated, linguistically motivated features.

The next chapter builds on the model described in Section 2.3 to build a basic model of ASR using CRFs for phone recognition. The chapters that follow continue to build on this framework to provide methods for more complex ASR tasks using CRFs.

CHAPTER 3: PILOT STUDY – PHONETIC RECOGNITION

This chapter presents a set of experiments that are a first exploration of the use of a CRF to integrate phone and phonological attribute information for ASR. The work in this chapter focuses on the use of CRF models for phonetic recognition using discriminative classifier outputs as input observations as well as comparing the effectiveness of these models to HMM-based models for phonetic recognition over the same feature sets. The experiments outlined in this chapter are the first proof-of-concept steps towards creating a CRF-based word recognition system, and as such the pilot system described herein is used as a base system to be expanded upon for CRF experiments in succeeding chapters.[5]

The outline of this chapter is as follows: Section 3.1 provides an overview of the experimental phone recognition task, as well as a brief description of the baseline systems used for comparison purposes. Section 3.2 discusses the structure of a CRF phone recognition that uses discriminative phone classifier output as input features. Section 3.3 describes a set of discriminative phonological attribute classifiers, as well as how the previously discussed phone classifier-based CRF model can be altered to accept these features for phone recognition. Section 3.4 examines how performance of these CRF systems can be improved through the addition of realignment and re-estimation to the training process, while Section 3.5 examines the performance improvements gained by using both the phone classifier and

[5]The work presented in this chapter was previously published in [43], [44], [45] and [46]

phonological attribute classifier outputs in a CRF system. Finally, in Section 3.6 the two-pass L-BGFS training method is compared with the Stochastic Gradient Descent training method to show that both methods achieve similar performance in this task.

3.1 Experimental Overview

These initial experiments perform the task of phonetic recognition using the TIMIT Acoustic Phonetic corpus [17]. TIMIT is a corpus of read, spoken English, collected from 8 different dialect regions in the United States. The corpus contains utterances from 630 different speakers, and is annotated with time markings for both word and phone boundaries, making it a corpus often used for phone recognition/classification experiments. Three types of utterances are used in the TIMIT corpus: *dialect* sentences which were spoken by all speakers across all dialects meant to bring out dialectal variation for further study, *compact* sentences designed to cover a wide variety of phone contexts, and *diverse* sentences designed to provide a larger diversity of phonetic contexts than the compact sentences. Following common practice, only the *compact* and *diverse* sentences were used in these experiments (as the *dialect* sentences are the same for all speakers, including them would bias the distribution of phones used in these sentences and lead to artificially inflated results).

The corpus is divided into a training set of 3697 utterances from 462 speakers, and a test set composed of 1344 utterances from 168 speakers. Speakers used for training are not used in testing as this would introduce a bias favorable to these speakers and could lead to artificially inflated results. These experiments use a standard partitioning of the test portion of the TIMIT corpus into a 24 speaker core test set (192 utterances) and a 50 speaker MIT development set (400 utterances) [19]. In addition, following Halberstadt and Glass in [23],

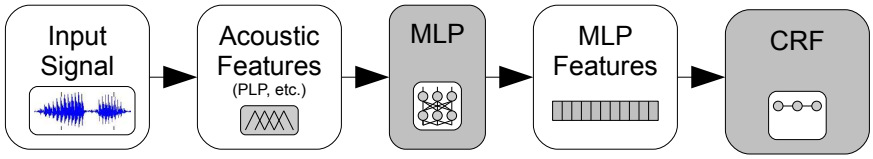

Figure 3.1: CRF phonetic recognition system overview

results are also reported for a larger test set of 118 speakers (944 utterances), containing the speakers in the core test set as well as the remaining speakers from the TIMIT test set that are not among the speakers in the development set. In this and future chapters, the TIMIT core test partition is referred to with the label *Core* and the larger test set referred to with the label *Test* by Halberstadt and Glass in [23] as *Enhanced*.

These experiments are performed using the outputs of ANN MLP classifiers as inputs to the CRF models. A diagram detailing the flow of this CRF system based on MLP classifier features is given in Figure 3.1. First, frames of PLP acoustic features are derived from the speech data. These PLP acoustic features are fed into an MLP classifier to generate a vector of class posterior features. These class posterior features are then used as feature functions in a CRF model to produce frame-level phone label assignments. Specific details on how the class posterior features are used to construct are given in the relevant "Model Description" sections for each experiment discussed below.

Although exact details on the nature of the outputs of these classifiers are given in the sections where they are used below, each classifier is constructed and trained in a similar manner. Tools from the ICSI Quicknet neural networks toolkit [12] are used to extract 12th-order PLP cepstral coefficients, plus and energy coefficient, along with first and second

order deltas, providing a 39-dimensional vector for each frame of speech. These extracted PLP coefficients are used to train ICSI Quicknet MLP classifiers. The MLPs used here are all built with 1000 hidden units and are trained using a nine-frame window of PLP coefficients (resulting in a 351 node input layer). Training is performed on a random selection of 3327 utterances from 416 speakers taken from the training set across all dialect regions and the MLPs were trained to convergence on a cross-validation set of 369 utterances from 46 speakers taken from the training set but disjoint from the speakers used to train the MLPs (to prevent overconfidence on the cross-validation set). CRF classifiers are built using a modified version of the Java CRF toolkit [61] using the L-BGFS for gradient descent during training. The CRF models are trained on all 3696 utterances from the training set, and are trained to convergence on the 50 speaker, 400 utterance development set.

To measure the performance of the CRF on the phonetic recognition task, the results of the CRF are compared to the results obtained through the use of Tandem HMM baselines as described by Hermansky et al in [25]. A diagram of the flow of a Tandem HMM system is shown in Figure 3.2. As with the CRF model described above, PLP acoustic features are first extracted from the speech signal and passed through an MLP classifier to generate a vector of class features.

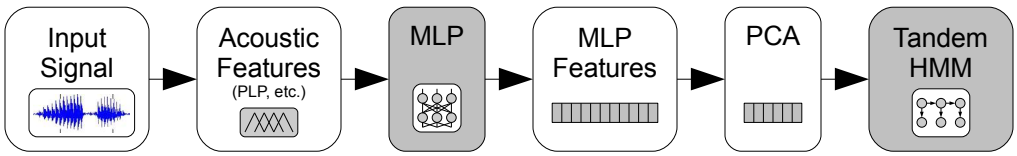

Figure 3.2: Tandem HMM system overview

As discussed by Hermansky et al, the skewed, non-Gaussian nature of the posterior vectors leads to poor performance if these vectors are used as-is in the Gaussian mixture models of the HMM. Typically in order to give the output of an MLP a probabilistic interpretation, the output layer is transformed via the application of a *softmax* function to the nodes of the output layer[54]. Equation 3.1 provides an example of the softmax function applied to an MLP neural network, where the term $output_i$ represents the i-th output of the MLP output layer and N is the size of the output layer. The application of this function generates a vector of length N values that sum to one.

$$y_i = \frac{\exp(output_i)}{\sum_{j=1}^{N} \exp(output_j)} \tag{3.1}$$

According to Hermansky et al, the performance of a Tandem system is significantly improved if the final non-linear transform step of the MLP is eliminated as features are generated. This has the effect of making the distribution of the MLP features a mean-shifted log transformation of the posterior features[6], as can be seen by taking the log of Equation 3.1. Instead of using the posterior value y_i for each class i, the value of $output_i$ is used directly as inputs to the HMM. In this work, following Hermansky et al, features generated in this manner will be referred to as *linear* MLP features, while features generated in the typical manner (i.e. through making the final application of the softmax function to the output layer) will be referred to as *posterior* features.

In addition, Hermansky et al reported an improvement in the accuracy of the Tandem HMM system when Principal Component Analysis (PCA) [10] was applied to the data to give a global decorrelation of individual features from one another. Following Hermansky

[6]Hermansky et al [25] also show that taking a log transformation of the posterior values can also improve performance, though the linear transformation gives a better result. This aspect of Tandem systems will be revisited in Chapter 4.

et al PCA is applied to these features via a Karhunen-Loeve (KL) transform before being used to train an HMM-based ASR system. All of the Tandem baselines used in this work are built using the HTK Toolkit [73].

Following the experimental design of Lee and Hon [36], for both the CRF and HMM based experiments system performance is evaluated using a reduced phoneme labeling for TIMIT of 39 possible phones instead of the full 61 phone labels. These mapping from 61 down to 39 labels consists of mapping together phone labels in TIMIT that do not provide confusions in the CMU dictionary pronunciations (e.g. stressed vs. unstressed vowels, stop closures) and so would cause not be a source of confusion in speech recognition. It is important to note that this mapping is only used in the evaluation of the CRF and Tandem HMM systems and that the MLP classifiers for phone classification generate outputs for all 61 possible phone labels. The Tandem HMM system results are reported for both full tied-state, word-internal triphone models as well as for monophone models. Triphone results are reported using a lattice-based language model that enforces triphone constraints and allows for biphone and monophone back-off but is not probabilistically weighted, as in the experiments that follow this lattice-based model gave accuracy results superior to a weighted bigram-based triphone lattice. Monophone results are reported using a bigram phone language model.

3.2 Phone Classifier Model

For these initial experiments, an MLP classifier was trained to predict *phone label* classes, and the outputs of this phone label MLP classifier were used in both a CRF phone recognition system and in a traditional Tandem HMM system. The phone classifier used in these experiments is a single MLP classifier, constructed and trained as described in

Section 3.1. The output layer of this phone classifier provides a vector of 61 outputs, each corresponding to one of the possible TIMIT phone class labels. The hand-transcribed phonetic transcriptions provided with the TIMIT corpus were used to generate frame-level label targets for the MLP classifiers. As described in Section 3.1, vectors of both *posterior* and *linear* features were generated by this MLP classifier for use in the CRF and Tandem HMM systems. Section 3.2.1 describes how the CRF model was constructed to use these MLP classifier outputs, while Section 3.2.2 provides the results of this initial experiment.

3.2.1 Model Description

Feature functions are defined for the CRF in line with the framework outlined by Equation 2.12 and Section 2.3.1. State feature function are created for each label/class pairing. For example, the following describes a feature function that ties together the output label $/t/$ with the phone classifier output for $/t/$:

$$f_{/t/,/t/}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} MLP_{PHN=/t/}(x_t), & \text{if } y_t = /t/ \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

These feature functions are defined for each label/class pairing, independent of the identity of the class or label. For example, the following state feature function is also defined for the output label $/t/$ based on the phone classifier output for the phone class $/d/$:

$$f_{/t/,/d/}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} MLP_{PHN=/t/}(x_t), & \text{if } y_t = /d/ \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

Building feature functions in this manner allows the CRF to obtain additional evidence when the MLP classifier makes an error – if the CRF system were only supplied with feature functions corresponding to the matching label assignments from the MLP, the CRF

would have less opportunity to be able to recover from the errors made by the MLP. The potential error illustrated in Equation 3.3 is one where the MLP has detected a high probability for the phone $/d/$ when the "true" label should be $/t/$. The realizations of $/t/$ and $/d/$ differ only in that $/d/$ is voiced while $/t/$ is not, so this may be an example of a frame where voicing from a preceding vowel has caused the $/t/$ to take on evidence of voicing. Allowing the CRF to see both outputs allows it more evidence to base its estimation from.

A different use for these "cross class" feature functions is illustrated in the feature function described in Equation 3.4. This is an example of an error that the classifier is unlikely to make – mistaking the $/d/$ of "DOG" for the $/ow/$ of "BOAT". Given a somewhat accurate classifier such misclassifications should be rare. However, that means that this feature function provides strong *negative* evidence for the label $/d/$ – when the $/ow/$ class has a high value, the true label is unlikely to be $/d/$.

$$f_{/ow/,/d/}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} MLP_{PHN=/ow/}(x_t), & \text{if } y_t = /d/ \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

In addition to the feature functions derived from the MLP classifier, *bias feature functions* are also implemented in this CRF formulation. There is exactly one bias state feature function for each label and one bias transition feature function for each label-label pair. These bias feature functions are non-zero if the label (or label pair) that they are defined for occur and are zero otherwise. For example, the following bias feature function is active for the phone label $/b/$:

$$f_{/b/,bias}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} 1, & \text{if } y_t = /b/ \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

Transition bias feature functions are defined in a similar manner: if the label-label transition described by the feature function occurs between two frames the feature function

fires with a value of one, otherwise the feature function has a value of zero. For example, the following transition feature function is active for transitions from $/b/$ to $/ah/$:

$$f_{/b/,/ah/}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} 1, & \text{if } y_t = /ah/ \ \& \ y_{t-1} = /b/ \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

Note that with the inclusion of bias features, in the absence of any evidence (i.e. all of the feature functions except the bias features evaluate to zero), the CRF model given in Equation 2.9 degenerates into a weighted sum of state bias functions and the transition bias functions. The weights on the state bias functions operate as a function of the unigram distribution of individual labels in the training set, while the weights on the transition bias functions operate as a function of the bigram distribution of label-label pairs[7]

As described in Section 3.1, the Tandem HMM baseline systems are trained using linear outputs of the MLP classifiers that have had a Karhunen-Loeve transform applied to them. In order to fully compare the results of the CRF to the Tandem baselines, a model using these transformed linear outputs as inputs to a CRF is also trained. For these models, the transition and bias feature functions are exactly as described above, while the state feature functions are defined using the transformed linear outputs instead of the softmax outputs. Because these inputs have been transformed through a principal components analysis, these CRFs lose the easy one-to-one correspondence between classifier outputs and labels. However, these additional experiments allow the elimination of differences in the inputs as a cause for differences in system performance. A feature function using these transformed values as inputs has the form:

[7]In addition to transition bias features, transition features that take values based on input features can also be crafted. In these experiments, however, only transition bias features are used. Transition features based on observations are discussed in later chapters.

$$f_{/b/,D1}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} KLT(MLP(x_t))_{D1}, & \text{if } y_t = /b/ \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

where $KLT(MLP(x_t))$ denotes the Karhunen-Loeve transformation of the MLP detector outputs obtained from the observation vector x_t and Dn denotes the nth dimension of that vector. In other words, the feature detector described in Equation (3.7) above returns the first dimension of the transformed feature vector when the current label is $/b/$ and returns a 0 when the current label is not $/b/$.

At training time, the values of all feature functions are easily determined from the training labels. At decoding time, all possible states and transitions are hypothesized and the most likely frame label sequence is found via the Viterbi algorithm as discussed in Section 2.3.3. Finally, consecutive frames that are assigned the same label in the most likely sequence are grouped together under a single label for evaluation of the accuracy of the labeled phone sequence. Note that this collapsing of frame labels can cause a phone deletion to occur in instances where the same phone appears twice in a row - such as in the pronunciation /hh iy iy t s/ for the phrase "he eats". This is a limitation of the single-state per phone label CRF model used in these experiments, but this limitation does not apply to the 3-state Tandem HMM systems used as baseline systems.[8]

To have enough data to train both the MLP classifier and the CRF, the TIMIT training data is used to first train the MLPs for classification. Once the MLPs have been sufficiently trained, they are applied to the training set to derive the phone class outputs used to train the CRF. (This process follows the procedure laid out in [25]).

The CRFs are trained via L-BFGS gradient descent, and the model described by these weight values is applied to the development set and the accuracy is computed. The weight

[8] Although the experiments in this chapter use only single-state models for CRFs, multi-state CRF models are examined in Chapter 5 to address this issue for a word-recognition framework.

values that give the highest accuracy on the development set are kept and used to determine the accuracy of the model on the core and enhanced TIMIT test sets.

3.2.2 Experimental Results

Model	Label Space	Feature Type	Number of Parameters	Core Accuracy	Enhanced Accuracy
PLP HMM (16mix)	triph.	PLP	1.3 million	67.4%	68.1%
Tandem Phone (16mix)	monoph.	linear+KLT	283,686	67.1%	67.9%
Tandem Phone (32mix)	monoph.	linear+KLT	567,078	68.3%	69.1%
Tandem Phone (16mix)	triph.	linear+KLT	1.7 million	69.3%	70.2%
CRF Phone	monoph.	posterior	5280	66.7%	68.1%
CRF Phone	monoph.	linear+KLT	5280	66.5%	67.5%

Table 3.1: *Phone classifier accuracy comparisons on TIMIT (61 inputs) for core test, and enhanced test sets. Significance at the $p \leq 0.05$ level is approximately 1.4%, and 0.6% percentage difference for these datasets, respectively.*

Table 3.1 shows a breakdown of recognition results for the CRF compared to a set of Tandem HMM baseline models. Accuracy results are reported for the 24 speaker partition of the TIMIT training set described in Section 3.1. Two different Tandem baseline models are examined for comparison purposes – a model trained for phone recognition using only monophone labels and a model trained using triphone labels. As with the CRF, the Tandem models are tuned using the development set and keeping the model parameters that provide the best development set performance. The best performance for the monophone model is achieved with a 32 Gaussian per state model while the best performance for the triphone model is achieved with a 16 Gaussian per state model. In addition, a 16 Gaussian per state monophone model which achieved the closest accuracy to the phone classifier CRF is also included for comparison purposes.

While the accuracy results of this phone classifier CRF does not meet the accuracy results of the best Tandem baseline models, its accuracy does approach the accuracy of the 16 Gaussian per state Tandem model. The difference in performance between the models is not significant ($p \geq 0.05$).[9] It is noteworthy that the CRF achieves this result with almost two orders of magnitude fewer parameters than the Tandem system, though the Tandem system is still able to achieve a better result by using additional parameters.

It is also worth noting that there is no significant difference between performance of the CRF trained using the posterior MLP outputs and the CRF trained using the linear, transformed MLP outputs. In fact, the performance of the system using the transformed linear outputs is marginally worse than the performance of the system using the posterior outputs, so the comparably better performance of the Tandem system cannot be attributed to the difference in the inputs.

3.3 Phonological Attribute Classifier Model

As a second experiment, the use of the outputs of a set of phonological attribute classifiers based on the attributes of the IPA phonetics chart as inputs to a CRF was investigated. In accordance with previous work done in the area of phonological feature extraction (see Section 2.2), multi-valued phonological attributes are extracted through a bank of MLP ANNs. The breakdown of these attributes expands on work performed by Rajamanohar and Fosler-Lussier in [57], and a complete inventory of the phonological attributes used for these experiments is outlined in Table 3.2. For each attribute category, a single n-ary MLP network is trained to detect the attributes in that category. For example, the MLP for the voicing attribute is trained with 3 possible output classes – voiced, unvoiced, and not

[9] All significance tests in this dissertation are reported using a one-tailed Z-test.

applicable. The outputs of these MLPs are then concatenated together into a single feature vector of 44 features for use in the CRF and Tandem HMM systems.

Table 3.2: *Phonological attributes extracted.*

Class	Output Attributes
SONORITY	vowel, obstruent, sonorant, syllabic, silence
VOICE	voiced, unvoiced, n/a
MANNER	fricative, stop, closure, flap, nasal, approximate, nasalflap, n/a
PLACE	labial, dental, alveolar, palatal, velar, glottal, lateral, rhotic, n/a
HEIGHT	high, mid, low, lowhigh, midhigh, n/a
FRONT	front, back, central, backfront, n/a
ROUND	round, nonround, roundnonround, nonroundround, n/a
TENSE	tense, lax, n/a

The labeling of phonological attributes is obtained in a straight-forward manner. Each hand-transcribed phone in the TIMIT phoneset is mapped to a vector of eight values that correspond to its canonical description as a bundle of attributes. Each phonological attribute classifier is then trained using these labels as the hard targets of the classifier. A breakdown of the mapping used for each phone label in the TIMIT phoneset can be found in Appendix A.

3.3.1 Model Description

The feature functions for the phonological attribute class CRF are constructed almost exactly as the feature functions for the phone class CRF above. State bias and transition bias functions between the two models are defined identically. State feature functions are defined using the label/phonological attribute pairs in a manner similar to how feature functions are defined in the phone classifier model described above. For example, the following

state feature function implements the feature function as described by Equation (3.2) to link the output label $/b/$ with the output of the VOICING attribute classifier for voiced speech:

$$f_{/b/,voi}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} MLP_{VOICE=voi}(x_t), & \text{if } y_t = /b/ \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$

where $MLP_{VOICE=voi}(x_t)$ designates the value of the voicing classifier for voiced speech on the frame x_t.

As with the phone classifier model above, state feature functions are defined for all possible label/attribute pairings, not just canonical attributes for the label. For example, in addition to the state feature function above the model also defines a state feature function that ties the phone label $/b/$ to the output of the VOICING attribute classifier for unvoiced speech:

$$f_{/b/,unvoi}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} MLP_{VOICE=unvoi}(x_t), & \text{if } y_t = /b/ \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

where $MLP_{VOICE=unvoi}(x_t)$ designates the value of the voicing classifier for unvoiced speech on the frame x_t. The state and transition bias features are defined in the same manner as in the system using phone classifier inputs described in the previous section.

Training and evaluation of the phonological attribute classifier CRF are performed in exactly the same manner as training and evaluation of the phone classifier CRF described above. As with the phone classifier CRFs, two different CRFs were trained for the phonological attribute classifier outputs – one model using the softmax posterior outputs, and one using the linear outputs transformed through the Karhunen-Loeve transform to compare to a similarly trained Tandem system.

Model	Label Space	Feature Type	Number of Parameters	Core Accuracy	Enhanced Accuracy
PLP HMM (16mix)	triph.	PLP	1.3 million	67.4%	68.1%
Tandem Ph. Att. (16mix)	monoph.	linear+KLT	205,350	66.6%	67.2%
Tandem Ph. Att. (32mix)	monoph.	linear+KLT	410,406	68.1%	68.6%
Tandem Ph. Att. (16mix)	triph.	linear+KLT	1.3 million	68.5%	69.3%
CRF Ph. Attr.	monoph.	posteriors	4464	65.8%	66.6%
CRF Ph. Attr.	monoph.	linear+KLT	4464	66.9%	67.5%
CRF Ph. Attr.	monoph.	linear	4464	65.8%	66.4%

Table 3.3: *Phonological Attribute classifier accuracy comparisons (44 inputs) for core test, and enhanced test sets. Significance at the $p \leq 0.05$ level is approximately 1.4%, and 0.6% percentage difference for these datasets, respectively.*

3.3.2 Experimental Results

Table 3.3 shows a breakdown of recognition results for the CRF compared to the comparable set of Tandem HMM baseline models. Again, results for a Tandem model trained for phone recognition using only monophone labels and a model trained using triphone labels are both shown for comparison purposes. The best performance for the monophone model is achieved with a 32 Gaussian per state model while the best performance for the triphone model is achieved with a 16 Gaussian per state model. Once again a 16 Gaussian per state monophone model which achieved the closest accuracy to the phone classifier CRF for comparison purposes.

Unlike the phone classifier CRFs, the phonological attribute CRF trained on the transformed linear MLP classifier outputs shows a substantial and significant ($p \leq 0.05$) improvement in accuracy over the CRF trained using the softmax posterior classifier outputs. To examine whether this improvement was achieved due to the linearization of the outputs or due to the application of principal components analysis, a third CRF was trained on

just the linear outputs of the MLP classifier without the application of the Karhunen-Loeve transform. As shown in Table 3.3, the CRF trained on just the linear outputs of the MLP classifiers achieved a result comparable to that of the CRF trained on the softmax outputs, indicating that the application of the KL transform is an important factor to improving recognition over the linear input features.

As with the phone classifier CRFs, the phonological attribute classifier CRFs do not achieve results comparable to the best results achieved by the Tandem models. However, as with the phone classifier CRFs one of the phonological attribute models does achieve a result comparable to a Tandem model with a much smaller number of parameters than the comparable Tandem model. Table 3.3 shows that the results achieved by the CRF trained on transformed, linear outputs of the MLP classifiers and the 16 Gaussian per state monophone Tandem model achieve comparable performance, but the CRF achieves this performance with substantially fewer parameters.

While neither basic CRF system achieves the accuracy of the 16 Gaussian triphone Tandem model, it is important to note some differences that the Tandem model has from the CRF that may be advantageous. Besides the obvious advantage of explicit triphone context in the labeling, the Tandem model explicitly models a three-state model for each phone label – the CRF makes no attempt to explicitly model different portions of a phone in a different manner. All phones in the CRF are modeled with the equivalent of a single state.

The second advantage that the Tandem system has over the CRF system lies in its training process. The Tandem system makes use of EM training, which allows for a probabilistic assignment of phone labels during the training stage. In contrast, the CRF system shown

here is trained only on fixed labels derived from the TIMIT training set. One approach to overcoming this disadvantage is addressed in the section that follows.

3.4 Viterbi Realignment Training

As discussed in the previous section, the requirement that the CRF have a fixed frame-level assignment of phone labels during training puts it at a disadvantage to the EM-trained HMM Tandem system, which allows for a probabilistic assignment of labels at training time. To compensate for this, the use of Viterbi realignment training for a CRF system was explored.

The training procedure is changed as follows: A CRF is trained as previously outlined. Then, using the weights derived from this CRF the training labels are realigned using a best-path Viterbi forced alignment. The weights used for this realignment are then used as initial seed weights for a new set of training iterations of the CRF. Again, this training stops when the accuracy of the model applied to the development test set stops improving. Although this training process can be repeated with a second pass of realignment and a second pass of retraining, in these experiments no additional improvement was gained through a second pass of realignment training. As such, results are reported here using only a single pass of Viterbi alignment training.

The results are shown in Table 3.4 (for phone classifier inputs) and in Table 3.5 (for phonological attribute classifier inputs). Results for the Tandem system trained with 16 Gaussian and triphone labels are included from Table 3.1 and Table 3.3 for comparison purposes. The results for the CRF trained on phone class posteriors and realigned are insignificantly better than those of the 16 Gaussian triphone Tandem system trained on phone classifier outputs. Likewise, the results for the CRF trained on the transformed,

linear phonological attribute classifier outputs are insignificantly better than those of the 16 Gaussian per state triphone Tandem system trained on phonological attribute classifier outputs. In both cases, the CRF achieves this result with substantially fewer parameters than the comparable Tandem system.

Table 3.5 also includes results for the CRF system trained on the linear outputs of the phonological attribute detectors without the application of the KL transform. This result, showing a worse performance for the CRF system than the Tandem system even after realignment, indicates that the gain in the performance of the CRF system using the linear, transformed outputs comes via the transformation of the outputs and is not strictly due to the linear outputs themselves.

One question that arises is whether an application of the KL transform might improve the results of the posterior outputs as well as it does the linear outputs. There is good reason to suspect that this would not be the case – principal component analysis techniques like the KL transform project data from an initial coordinate system into a new coordinate system where the dimensions of the new coordinate system correspond with the variance of the initial data. The dimension with the highest variance becomes the first (or principal) dimension of the transformed data, and the remaining dimensions are determined in descending order of variance in the initial data [10].

However, the variance on the initial posterior features is necessarily going to be diminished when compared to the variance of the linear outputs. Recall the form of the softmax function from Equation 3.1. When one of the outputs of the neural network dominates over the others, the application of the softmax function pushes that value closer to one and the remaining values close to zero. As such, most of the features in the posterior vectors will be close to zero, with one feature in each of the phone class vectors taking a value no larger

than one or 8 features in the posterior phonological attribute vectors doing likewise. An experiment was carried out to examine this effect on the phonological attribute posterior vectors. The low variance of these vectors caused a drop in the dimensionality of the data following the KL transform – only 42 dimensions were available for examination instead of the original 44. A CRF was trained over this reduced set of features, and the difference in performance between it and a system trained over the posterior features directly was statistically insignificant – the application of the KL transform did not improve performance the way it did for the linear features, suggesting that the variance in the linear features may be an important aspect for use in these systems.

Model	Label Space	Feature Type	Core Accuracy	Enhanced Accuracy
PLP HMM (16 mix)	triph.	PLP	67.4%	68.1%
Tandem Phone (16mix)	triphone	linear+KLT	69.3%	70.2%
CRF Phone	monophone	posteriors	69.3%	70.4%
CRF Phone	monophone	linear+KLT	68.5%	69.2%

Table 3.4: *TIMIT Phone classifier accuracy comparisons after realignment (61 inputs) for core test, and enhanced test sets. Significance at the $p \leq 0.05$ level is approximately 1.4%, and 0.6% percentage difference for these datasets, respectively.*

Table 3.6 and Table 3.7 show a breakdown of the overall performance of each CRF, both re-aligned and without realignment. It is readily apparent where the increase in accuracy comes from. The number of correct labels hypothesized by the CRFs have increased by anywhere from 3-3.75%. Simultaneously, we see that the number of insertions have almost doubled in number – leading to accuracy gains of only 1.5-2.3% for the individual CRFs.

Model	Label Space	Feature Type	Core Accuracy	Enhanced Accuracy
PLP HMM (16mix)	triph.	PLP	67.4%	68.1%
Tandem Ph. Attr. (16mix)	triphone	linear+KLT	68.5%	69.3%
CRF Ph. Attr.	monophone	posteriors	67.7%	68.5%
CRF Ph. Attr.	monophone	linear+KLT	69.2%	69.8%
CRF Ph. Attr.	monophone	linear-only	66.6%	67.1%

Table 3.5: *TIMIT Phonological attribute classifier accuracy comparisons after realignment (44 inputs) for core test, and enhanced test sets. Significance at the $p \leq 0.05$ level is approximately 1.4%, and 0.6% percentage difference for these datasets, respectively.*

Table 3.6: *Phone classifier model detail comparisons before and after realignment (61 inputs).*

Model	Feature Type	Enhanced Accuracy	Correct	Dels	Inserts	Subst
Tandem Phone (16mix)	linear+KLT	70.2%	24559	1835	1964	5786
CRF Phone	posteriors	68.1%	22421	4602	503	5157
CRF Phone (realigned)	posteriors	70.4%	23606	2976	950	5598
CRF Phone	linear+KLT	67.5%	22322	4228	607	5630
CRF Phone (realigned)	linear+KLT	69.2%	23355	2880	1084	5945

Comparing the CRFs to their counterpart Tandem HMM models show similar results for each model. The CRFs have between 2-4% fewer correct labels than the corresponding Tandem model, and the CRFs continue to have a higher number of deleted phones than the corresponding Tandem model. However, the CRFs all continue to show fewer insertions than the corresponding Tandem model – in all cases but one the CRFs have less than half the number of insertions of the similar Tandem model. It is the overall gain in correctness

Table 3.7: *Phonological attribute model detail comparisons before and after realignment (44 inputs).*

Model	Feature Type	Enhanced Accuracy	Correct	Dels	Inserts	Subst
Tandem Ph. Attr. (16mix)	linear+KLT	69.3%	24167	1874	1876	6139
CRF Ph. Attr.	posteriors	66.6%	21938	4649	507	5593
CRF Ph. Attr. (realigned)	posteriors	68.5%	22957	3163	906	6060
CRF Ph. Attr.	linear+KLT	67.5%	22300	4248	575	5632
CRF Ph. Attr. (realigned)	linear+KLT	69.8%	23506	2804	1041	5870

that the realignment allows combined with the continued comparable sparsity of insertions that allows the CRF to achieve an accuracy result comparable to the Tandem system.

Looking more closely at the results of the two phonological attribute-based CRFs in Table 3.7, it is clear that the gains in performance made by the linear-transformed outputs over the posterior outputs are attributable to both a substantial decrease in overall deletions as well as a smaller reduction in the number of substitutions. This comes at a cost of a small increase in the number of overall insertions. The improvements in deletions and substitutions is spread over all phones – no single label or group of labels improves at the expense of the others. Likewise, the increase in insertions is spread over all phones.

The results of the posterior-trained phone class CRF are significantly better on the Enhanced test set ($p \leq 0.05$) than the results of a CRF trained on the transformed linear outputs of the phone classifier. It is interesting to note that the phone class posterior outputs are highly correlated with each other, yet decorrelation provides no increase in performance. This is another piece of evidence that suggests that the improvement in the phonological attribute classifier space may not be coming due to the decorrelation of the inputs (as appears to be the case with the HMM model), but instead may be due to the transformation

of the space into the variance space of the outputs. It is also noteworthy that the difference in accuracy between the best phone classifier CRF and the best phonological attribute CRF is not significant.

3.5 Feature Combinations

A key strength of the CRF model is said to lie in its ability to incorporate many different attributes of the observed sequence without regard for possible correlations. To examine this idea, a CRF system that was trained on an input set that makes use of both the phonological and phone class attributes simultaneously to see if an increase in performance could be obtained with information that is supposedly redundant.

The results of these experiments are shown in Table 3.8. Results for a Tandem system supplied with linear MLP outputs and a K-L transform applied to the combined outputs are also reported. Two results are reported for the CRFs – the first with phone class and phonological attribute class outputs as posteriors, and the second with the phone class outputs as posteriors and the phonological attribute class outputs as linear, K-L transformed outputs (i.e. the best results from the previous section). Both CRFs are trained using the Viterbi realignment training as outlined in the previous section.

The performance of the Tandem system trained with all 105 attributes is not significantly different than the performance of the Tandem system trained only on phone classes. Conversely, the performance of the CRF system trained on the posterior phone classes and the transformed linear phonological attributes is not only significantly better than that of the Tandem system, it is also significantly ($p \leq 0.05$) better than that of the CRF trained on only the phone classifier outputs. The improvement in performance for the CRF trained on all 105 posterior outputs over the CRF trained on only the 61 phone class outputs is not

Model	Feature Type	No. of Inputs	Number of Parameters	Core Accuracy	Enhanced Accuracy
Tandem Phone [16mix]	linear+KLT	61	1.7 million	69.3%	70.2%
Tandem All [16mix]	linear+KLT	105	2.8 million	69.9%	70.2%
CRF Phone (61 inputs)	posteriors	61	5280	69.3%	70.4%
CRF All (105 inputs)	posterior	105	7392	69.9%	71.0%
CRF All (105 inputs)	post.&lin+KLT	105	7392	70.7%	71.5%

Table 3.8: *Phone accuracy comparisons with all attributes for core test and enhanced test sets. Significance at the $p \leq 0.05$ level is approximately 1.4%, and 0.6% percentage difference for these datasets, respectively.*

significant on the core test set, but is significant on the larger enhanced test set. Note also that the result is obtained with only a fraction of the parameters needed to model all 105 attributes in the Tandem system.

Comparing the results of the CRF trained with all 105 attributes against the CRF trained only on 61 phone classes shows an overall improvement in the correct labeling of almost all phones. Table 3.9 shows a comparison of the CRF using only posterior phone class outputs to the model using both the posterior phone class outputs and the transformed, linear phonological attribute class outputs. Using all 105 attributes substantially improves the overall correctness of the model by 1.4%, mainly through a large reduction in the number of deleted phones and a minimal reduction the number of substitutions. This comes at the expense of a small increase in the number of insertions for the model that reduces the overall improvement in accuracy to roughly 1%.

Another interesting question is why the CRF is able to compete using a one-state model with a three-state triphone model. One possibility is that the MLP classifiers, which incorporate a 9-frame context window, obviates the need for the three state model; another

Model	Feature Type	Enhanced Accuracy	Correct	Dels	Inserts	Subst
CRF Phone	posteriors	70.4%	23606	2976	950	5598
CRF All	posteriors & linear+KL	71.5%	24058	2570	1054	5552

Table 3.9: *TIMIT Phone recognition comparisons phone classifier only vs. phone classifier + phonological attributes.*

possibility is that the CRF's additional degrees of freedom in its exponential model can somehow compensate better for the diverse input. The truth seems to be a combination of these reasons. A monophone HTK system was trained on the phonological attribute data using only one state per phone; the resulting system is roughly 6% (absolute) less accurate than the 3-state system. Conversely, a PLP-based 1-state monophone HTK system is around 11% (absolute) less accurate than a corresponding 3-state system. These results indicate that the windowed posterior estimates from the MLP do compensate to some degree for an impoverished state space in the statistical model; however, the differential between the one and three state systems indicates that this compensation is incomplete, suggesting that the CRF is using the posterior estimates more efficiently than an HMM in a one-state model. In Chapter 5 the CRF model used here is extended from a single state model to a 3-state model for word recognition, and the 3-state monophone CRF model significantly outperforms the 3-state monophone Tandem HMM model, which is additional evidence for the suggestion that the CRF is making better use of this evidence than the comparable HMM.

3.6 Stochastic Gradient Training

The work performed in [20] showed that the stochastic gradient descent (SGD) method of training CRFs gave improved performance and shorter training times than the quasi-Newton L-BFGS method of gradient descent. SGD training was implemented following this work as outlined in the previous section, and the results were compared to the results obtained through L-BFGS.

Model	Feature Type	Phone Accuracy	Correct	Dels	Inserts	Subst
Tandem Phone	linear+KLT	70.2%	24559	1835	1964	5786
CRF Phone (L-BFGS)	posteriors	70.4%	23606	2976	950	5598
CRF Phone (SGD)	posteriors	70.7%	23667	3151	913	5362

Table 3.10: *Phone accuracy comparisons SGD vs. L-BFGS training for Phone Classifiers (61 inputs) for enhanced test set. Significance at the $p \leq 0.05$ level is approximately 0.6% percentage difference for this dataset.*

Model	Feature Type	Phone Accuracy	Correct	Dels	Inserts	Subst
Tandem Phono.	linear+KLT	69.1%	24167	1874	1876	6139
CRF Phono. (L-BFGS)	posteriors	67.8%	22957	3163	906	6060
CRF Phono. (SGD)	posteriors	68.0%	23750	2247	1810	6183

Table 3.11: *Phone accuracy comparisons SGD vs. L-BFGS training for Phonological Attribute classifiers (44 inputs) for enhanced test set. Significance at the $p \leq 0.05$ level is approximately 0.6% percentage difference for this dataset.*

Table 3.10 shows the results of the SGD training compared to Tandem HMM and CRF L-BFGS gradient descent training with Viterbi realignment for phone classifier inputs,

while Table 3.11 shows the same for phonological attribute classifier inputs. The difference in the accuracy results between the two different CRF models is not statistically significant for either set of features. The superior performance of the SGD training paradigm comes in the time it takes to train the model – training for the L-BFGS model took close to 1000 iterations through the training set over multiple days to achieve the reported results, and included a realignment pass. Training for the SGD model took only 15 iterations for the phonological feature CRF and 22 iterations for the phone class CRF – each completing their training in a matter of hours instead of days. These results support the findings of Gunawardana et al and show that they apply to phone recognition as well as classification.

One observation to note in both Table 3.10 and Table 3.11 that although the differences between the two systems are statistically insignificant, the character of the results they give are not exactly the same. In both cases the system trained via SGD shows a larger number of correct phone labels than the comparable L-BFGS trained system (though the Tandem system achieves a higher correctness than either CRF system shown here). The two systems also show a difference in the number of insertions, deletions and substitutions. Although the two methods provide models that are substantially similar to one another, they are not providing exactly the same results.

Table 3.12 shows a comparison of an L-BFGS trained system and an SGD trained system over posterior features for a mix of phone classes and phonological attribute classes. Again the difference between the SGD trained system and the L-BFGS system is statistically insignificant, but in this case the SGD trained system achieves a lower overall accuracy than the L-BFGS system rather than a slightly higher accuracy. In this case, the penalty to the accuracy comes with the increased number of insertions in the model – the SGD trained system shows a reduced number of deletions, and substitutions as well as more

correct phone classifications compared to the L-BFGS trained system, but the substantial increase in insertions negatively impacts the overall accuracy of the model.

Model	Feature Type	Phone Accuracy	Correct	Dels	Inserts	Subst
Tandem Phn+Phono.	lin+KLT	70.2%	24792	1601	2205	5787
CRF Phn+Phono.(L-BFGS)	posteriors	71.0%	23761	2805	930	5614
CRF Phn+Phono.(SGD)	posteriors	70.4%	24612	2001	1964	5567

Table 3.12: *Phone accuracy comparisons SGD vs. L-BFGS training for Phone Classifiers and Phonological Attribute classifiers (105 inputs) for enhanced test set. Significance at the $p \leq 0.05$ level is approximately 0.6% percentage difference for this dataset.*

These results for the SGD training do not make use of the Viterbi realignment training method. Despite its effectiveness in improving the results of L-BFGS training, every test attempting to combine Viterbi realignment with SGD training has yielded no improvement in the final model (and in some cases even yielded an insignificant decrease in accuracy). This is possibly due to the use of parameter averaging in the SGD training scheme – when SGD without parameter averaging is used, the use of a Viterbi realignment pass does improve the results. However the final results of a system trained without parameter averaging – even including a pass of Viterbi realignment - have in all tests been significantly lower than the results of the same system trained with parameter averaging. As such, all results in this dissertation that use the SGD training method are reported with parameter averaging and no Viterbi realignment.

3.7 Summary

This chapter has presented a pilot study into feature-based phone recognition using the model of Conditional Random Fields. These experiments have shown that a basic, single-state, monophone context CRF model can be used to combine a set of phonological feature streams and achieve phonetic recognition results superior to that of a monophone context, single Gaussian HMM model and comparable to that of a triphone context, multiple Gaussian mixture model HMM system trained on the same set of features. They have also shown that the CRF model can achieve these results with not only a much smaller context, but also with a much smaller set of parameters to model the space.

Additionally these experiments have shown that features that are highly correlated (such as phonological features and phone classes) can be added to a CRF system in a straightforward manner and give significant improvements in phone recognition performance. In these experiments, these improvements come not at the expense of one set of phones over another set, but instead by raising the overall performance of almost all of the phones in the test set. While adding features to a comparable HMM system does improve correct labellings, it comes at the expense of many spurious insertions that affect overall accuracy. In contrast, the CRF model shows improvement in overall recognition accuracy, with an increase in correct labels and a reduction in insertions, deletions and substitutions. It is worth noting that none of the models in these experiments yet approach the best results for an HMM system of roughly 75% for the task of phone recognition on the Core test set ([9],[22]) and of 79.04% on the full TIMIT test set ([65]). The results here are designed to show a comparative assessment between the two models on the same set of discriminatively trained inputs.

This pilot study supports the idea that the CRF model holds promise for ASR. But in order to benefit from these results, CRF models need to be able to do more than just phone recognition. In the next two chapters methods of extending the pilot systems outlined here from phone recognition to word recognition are proposed and analyzed.

CHAPTER 4: WORD RECOGNITION VIA THE USE OF CRF FEATURES IN HMMS

The work discussed in the previous chapter shows that a CRF model can obtain better results for phone recognition than a similarly trained HMM model. However, to be of use in ASR systems these models need to be able to move beyond phone recognition and perform word recognition. This chapter and the chapter that follows discuss two different approaches to this problem. One potential approach, outlined in this chapter, is to take inspiration from "Tandem"-style HMMs as described in Chapter 3 and use a CRF model to produce output suitable for use as input to an existing HMM-based system. This combined CRF-Tandem HMM (or "Crandem") system is able to benefit from existing ASR models and technology for word recognition while incorporating the superior phone recognition results of the CRF model.[10]

The outline for the rest of this chapter is as follows: Section 4.1 quickly reviews the structure of the Tandem system and describes how a trained CRF model can be used to generate features for a modified Tandem system (dubbed a "Crandem" system). Section 4.2 provides an overview of our experimental pilot system for phone recognition over the TIMIT corpus, as well as a discussion of the results of the pilot system. Section 4.3 gives a description of our experimental word-recognition system as well as experimental results and analysis from the Crandem system.

[10]The work discussed in this chapter was previously published in [13] and [47].

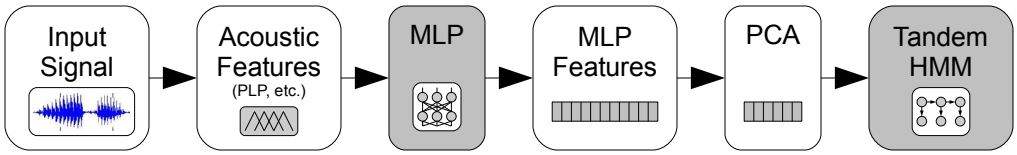

Figure 4.1: Tandem system overview

4.1 Crandem System Outline

As described in Chapter 3, a Tandem HMM system is a convenient method for integrating the output of a discriminative classifier into an HMM-based speech recognition system. Figure 4.1 reiterates the illustration of the flow of a Tandem HMM system. In a Tandem system acoustic input is transformed from an acoustic frequency representation (e.g. PLP coefficients, MFCCs, etc.) into a discriminative representation of the signal via a transformation function. This transformation function is usually a MLP classifier trained to discriminate among phone classes (as by Hermansky et al in [25]) but other models (such as phonological feature classifiers by Launay et al in [34]) have also been explored. As described by Hermansky et al in [25] and discussed in Section 3.1, the outputs of the MLP neural network are transformed either by taking a log transformation of the outputs or by omitting the final application of the *softmax* function to the output layer. These transformed outputs are decorrelated via an application of Principal Components Analysis (PCA) - specifically the Karhunen-Loève (KL) transform - and then used to build a likelihood-based HMM system.

In order to take advantage of the improved phone recognition of the CRF system described in Chapter 3, the extension of the Tandem model proposed in this chapter places a

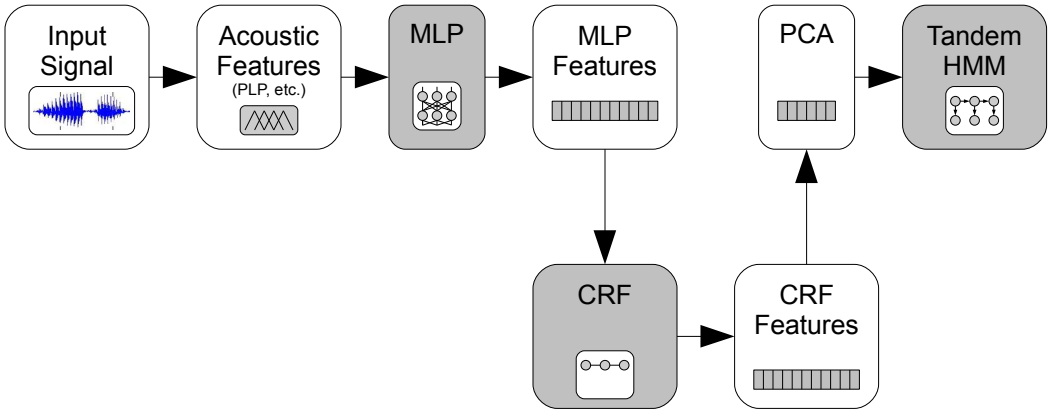

Figure 4.2: Tandem system modified for CRF Features (Crandem)

discriminative CRF classifier between the MLPs and the HMM system. Figure 4.2 shows a diagram of the flow of the proposed Crandem system. Unlike a MLP classifier, which can take in a single frame of speech and output a probability of a phone label given that frame of speech, a CRF classifier evaluates the probability of an entire sequence of phone labels given the entire sequence of input speech features to provide a global estimate for the probability of an entire utterance. A mismatch of output from the CRF and desired input for the HMM exists: in order to use the results of a CRF in a system such as this the CRF needs to be modified to provide frame level local estimates for phone classes rather than the single global estimate for the whole sequence.

Previous discussion of the training of CRFs in Section 2.3.1 suggests a solution: the CRF training regime already makes use of a variant of the forward-backward algorithm that computes local posterior estimates of a set of labels based on the global CRF model. While this algorithm was initially derived to compute local posterior probabilities for training purposes, it can easily be repurposed to generate frame level posterior probability estimations

for use as HMM inputs. Equation 4.1 reiterates the form of the CRF model (reproduced from Equation 2.11):

$$P(\mathbf{Y}|\mathbf{X}) = \frac{\exp \sum_t (\sum_i \lambda_i s_i(y_t, \mathbf{x}, t) + \sum_j \mu_j f_j(y_{t-1}, y_t, \mathbf{x}, t)}{Z(\mathbf{x})} \tag{4.1}$$

Where each s_i (with associated weight λ_i) is a state feature function that associates an input vector X with a phone label y. Additionally, each f_j (with associated weight μ_j) is a transition feature function that associates the vector X with a phone label transition between a pair of states y and y', and the $Z(X)$ term is a normalization constant over all possible paths over the input X. As shown in [62] and discussed in Section 2.3.2, this can be reformulated as a version of the forward-backward algorithm:

$$P(y_{i,t}|X) = \frac{\alpha_{i,t}\beta_{i,t}}{Z(X)}, Z(X) = \sum_j \alpha_{j,t}\beta_{j,t} \tag{4.2}$$

where α and β are defined as a collection of potentials leading up to a particular time step (α) and from that time step to the end of the utterance (β) similar to the alpha-beta recurrence in standard E-M training for HMMs. Using this recurrence, a CRF model trained for phone recognition can now be used to generate a vector of posterior probabilities suitable for use in a Tandem-like Crandem system.

As noted in the previous section, Tandem systems perform poorly when posteriors are used directly as input features and so the application of some transformation to the posterior outputs of the CRF is desirable. While the log transformation can be applied to the CRF posterior outputs directly (suitably flooring for $log(0)$), the $linearize$ transformation cannot as the CRF does not apply a softmax function to the frame-level outputs to get posteriors. However, an analogous transformation can be used if the application of the $Z(X)$ denominator term is omitted from the computation of the posteriors described in Equation

4.2. This transformation is designated $unnorm$ (for "unnormalized") in the experiments below. The KL-transform described above, on the other hand, is just as applicable to the transformed posteriors from the CRF as it is to the transformed outputs of the MLP and can be used in the same manner.

4.2 Experimental design: Phone Recognition Pilot

Before investigating this Crandem model for word recognition, a pilot system for phone recognition was first built and tested. The phone recognition pilot is an extension of the phone recognition CRF systems described in Chapter 3, and was used to determine if the ideas for extending the Tandem system to a Crandem system outlined in the previous section would be fruitful. The Crandem systems described here are built as modifications of the Tandem baselines used in Chapter 3.

4.2.1 Phone Posterior Inputs

For these experiments, new ICSI Quicknet MLPs [12] were trained over 39 dimensional PLP coefficients extracted from the TIMIT training set of 3697 utterance from 462 speakers. The training of these MLP networks follows the same protocol discussed in Chapter 3. The same division of 3327 utterances from 416 speakers was used for actual training of the networks and the same set of 369 utterances from 46 speakers was used for cross-validation to determine convergence. The MLP neural networks in this section were built using a larger 2000 unit hidden layer which provided improved accuracy for the classifiers.

New CRF phone recognition systems were built with these MLP outputs using the process described in Chapter 3. Frame level outputs from these CRFs were then acquired as described in the previous section and the Karhunen-Loève (KL) transform was applied to these frame level outputs to provide decorrelation. These frame level results were then

used as input features to train an HMM using the HTK toolkit [73]. As with the work in Chapter 3, these CRF models are one-state-per-phone monophone label systems.

In addition to the CRFs with only state features as described in Chapter 3, a second type of CRF was examined in this pilot – a CRF where transition features were used as well as state features. These transition features use the same MLP posterior outputs that the state features use, but in the transition features the MLP outputs are associated with a label to label transition pair rather than a single label. This allows for a small amount of context dependence with still using monophone CRF labels.[11] Expanding on the discussion of CRF transition functions provided in Chapter 2, a sample transition feature for this system is shown in Equation 4.3. This function ties the value of the MLP phone classifier output for the label $/ah/$ with a transition in the label sequence from $/d/$ to $/ah/$ – the function takes a non-zero value only when the MLP classifier provides a non-zero value for evidence of the phone $/d/$ at time t in the speech signal and the hypothesized label sequence $\mathbf{y}$ contains a transition from the phone label $/d/$ to the phone label $/ah/$ at time t.

$$f_{/d/,/d/,/ah/}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} MLP_{PHN=/d/}(x_t), & \text{if } y_{t-1} = /d/ \text{ and} \\ & y_t = /ah/ \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

As a second example, consider the feature function described in Equation 4.4. This example shows the case where the hypothesized label sequence contains a *self transition* at time t – the phone hypothesized at time $t-1$ remains the same as at time t. In this case, the function will take a non-zero value when the MLP classifier has a non-zero value for evidence of the phone $/ah/$ at time t in the speech signal and a transition from the phone label $/ah/$ to the same phone label $/ah/$ occurs in the label sequence $\mathbf{y}$.

[11] Feature functions designed specifically to find evidence of transitions, rather than re-using the same feature functions used for state feature functions, could possibly achieve better performance.

$$f_{/ah/,/ah/,/ah/}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} MLP_{PHN=/ah/}(x_t), & \text{if } y_{t-1} = /ah/ \text{ and} \\ & y_t = /ah/ \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

Finally, as with the state feature functions described in Chapter 3, the transition feature functions are not restricted to using MLP outputs that match the labels. Equation 4.5 shows an example of this kind of transition feature. This feature takes a non-zero value when the label sequence hypothesizes a transition from the phone $/d/$ to the phone $/ah/$ at time t and the MLP has a non-zero value for the phone class $/t/$ at time t. As with state feature functions, transition feature functions are crafted for all possible combinations of label pairs and MLP outputs to allow the CRF to gain evidence from misrecognitions by the MLP or contexts where pronunciation shifts occur.

$$f_{/t/,/d/,/ah/}(\mathbf{y}, \mathbf{x}, t) = \begin{cases} MLP_{PHN=/t/}(x_t), & \text{if } y_{t-1} = /d/ \text{ and} \\ & y_t = /ah/ \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

A number of baselines were used for comparison purposes. In addition to the phone recognition of the original CRF and a traditional 32 mixture, tied-triphone HMM system trained over the initial PLP cepstral coefficients, a 32 mixture, tied-triphone Tandem system was built. This system used the linearized, KL-transformed outputs of the MLP as inputs. In addition, since the CRFs trained over TIMIT produce a labeling over 48 possible phones (rather than the 61 phones that the MLPs provide), there was some possibility that there might be a gain due to the dimensionality reduction alone. A second Tandem system was trained by reducing the dimensionality of the MLPs after the KL-transform from 61 down to 48 dimensions for the Tandem system.

A final baseline arose from the question of how much is gained from the CRF as an aggregator of local posterior outputs. To this end, an MLP was trained over the same

	System	Dev	Core	Ext
	PLP HMM reference	69.7	67.4	68.1
1	Tandem (61 ftrs)	72.1	69.4	70.6
2	Tandem (48 ftrs)	72.6	69.6	70.8
3	CRF (state only)	71.1	68.9	69.9
4	CRF (state+trans)	71.4	69.5	70.7
5	MLP-Tandem	70.0	67.2	68.2
6	Crandem$_{\log}$ (state)	72.9	69.8	71.1
7	Crandem$_{\log}$ (state+trans)	73.1	70.5	71.7
8	Crandem$_{\text{unnorm}}$ (state)	73.1	70.1	71.2
9	Crandem$_{\text{unnorm}}$ (state+trans)	73.1	70.6	71.8

Table 4.1: *Phone class posterior results. Phone accuracies on TIMIT for development, core test, and extended test sets. Significance at the $p \leq 0.05$ level is approximately 0.9%, 1.4%, and 0.6% percentage difference for these datasets, respectively.*

data that the CRF was trained over (i.e. phone posterior outputs of an initial MLP) and another Tandem system (Tandem-MLP) was trained over the outputs of this MLP. Like all Tandem systems, the results reported here are from an HMM trained over the linearized, KL-transformed outputs of the MLP.

All of the HMM based system were tuned on the development set of 400 utterances from the TIMIT test set, as outlined by Halberstadt and Glass [23] and discussed in more detail in the previous chapter. Results are reported for the "core" set of 192 utterances as well as for the 944 remaining utterances neither in the "core" nor in the "development" partitions of the test set (here labeled "extended" or "ext").

Results from these initial phone recognition experiments are shown in Table 4.1. The differences in the various systems are not significant on the core test set (due to the small size of this set), so following common practice results are also reported on the extended test set. All measures of significance are reported using a one-tail Z test. As shown in Table 4.1,

System 3 (CRF state only) performs significantly worse than either of the Tandem baselines on the extended test set. This significance goes away with the addition of transition features in System 4 (CRF state+trans).

Note that in all cases the Crandem system performs better than the corresponding Tandem system and the corresponding CRF system in this task, though not all performances are significant. Specifically, the gain between Crandem System 6 (Crandem log) and Tandem System 2 (Tandem 48 ftrs) is not significant, nor is the gain between Crandem System 8 (Crandem unnorm) and either of the two Tandem Systems. In all other cases, however, the improvement in phone recognition between a Crandem system and its corresponding baseline Tandem system is significant, as is the performance gain in phone recognition between the Crandem system and its corresponding CRF system.

The gains in performance by the Crandem systems over the Tandem systems cannot be explained by dimensionality reduction alone. First, the performance gain between Tandem System 1 (61 ftrs) and Tandem System 2 (48 ftrs) is not significant. Second, two of the Crandem systems - Crandem System 7 and Crandem System 9 - significantly outperform the dimensionality reduced Tandem System 2. The gains in performance by the Crandem systems also cannot be explained as an effect of their input features - the MLP-Tandem system trained on the same inputs as the CRFs performs significantly worse than any other system in the list.

4.2.2 Phone Posterior and Phonological Posterior Inputs

Building on the first set of experiments, a second set of phone recognition experiments was performed. Rather than just using the phone class posteriors to build a CRF, these experiments extend the work performed in Chapter 3 and examine the use of CRF models

	System	Dev	Core	Ext
1	Tandem (105 ftrs)	72.2	69.7	70.9
2	Tandem (48 ftrs)	72.5	70.2	71.2
3	CRF (state only)	72.7	70.3	71.4
4	CRF (state+trans)	72.7	70.9	71.6
5	MLP-Tandem	71.4	69.4	70.8
6	$\text{Crandem}_{\text{log}}$ (state)	73.0	70.7	71.7
7	$\text{Crandem}_{\text{log}}$ (state+trans)	73.4	71.2	72.4
8	$\text{Crandem}_{\text{unnorm}}$ (state)	72.9	70.6	71.7
9	$\text{Crandem}_{\text{unnorm}}$ (state+trans)	73.4	70.8	72.4

Table 4.2: *Phone class and Phonological attribute class posterior results. Phone accuracies on TIMIT for development, core test, and extended test sets. Significance at the $p \leq 0.05$ level is approximately 0.9%, 1.4%, and 0.6% percentage difference for these datasets, respectively.*

that combine phone class posteriors and phonological attribute class posteriors for phone recognition. The same phone class posteriors from the previous section were combined with phonological attribute posteriors as described in Chapter 3. The exact same set of baseline systems – altered only to allow the use of both phone class outputs and phonological attribute class outputs as inputs – were built and compared.

Results for the second set of Crandem phone recognition experiments are shown in Table 4.2. Note that the pattern of results is similar to that of the phone class posteriors: The Crandem systems show an improvement over both the Tandem systems and the initial CRF, though in this case only the Crandem systems that include both state and transition features show significance in their improvement.

Note that in both Table 4.1 and Table 4.2 while the CRF trained with both state and transition features sees only insignificant gains in accuracy over the CRF trained using

state features alone, the Crandem systems show larger (and in the case of $\text{Crandem}_{\text{log}}$ significantly larger) gains in performance. This suggests that even redundant information on the transitions is benefiting the downstream processing in the Crandem system even if these benefits do not show up in the accuracy of the underlying CRF itself.

The CRF-based models all gain more benefit from the addition of phonological features than the comparable Tandem systems. While all of the systems show some improvement when phonological features are added, this improvement is not significant in any of the HMM-based systems. It is significant in all of the CRF and Crandem systems except for the $\text{Crandem}_{\text{unnorm}}$ systems. The CRF-based systems are consistently better able to bring together the redundant information provided by both the phone class posteriors and the phonological feature class posteriors.

Finally, in the literature it is common to combine Tandem features with traditional acoustic features to achieve better overall performance [75]. Table 4.3 shows the results of a system that appends the original PLP features with the best-performing Crandem system above – the $\text{Crandem}_{\text{log}}$ system using both state and transition features, with phone and phonological feature class inputs. Combining these features together shows a significant improvement over the original baseline PLP system as well as a smaller, but still significant, improvement over the Crandem system without the PLP features. This results suggests that the Crandem feature set – like the Tandem feature set – supplements the information provided by traditional acoustic features and can be useful as a supplemental set of features for enhancing the performance of a system.

System	Dev	Core	Ext
PLP HMM reference	69.7	67.4	68.1
Crandem$_{\log}$ (state+trans)	73.4	71.2	72.4
PLP + Crandem$_{\log}$ (state+trans, phone+phono)	74.3	71.8	73.3

Table 4.3: *Phone accuracy for TIMIT with an HMM system trained with PLP coefficients appended to System 7b (Crandem$_{\log}$ (state+trans) trained on 61 phone class and 44 phonological attribute posteriors).*

4.3 Experimental Design: Word Recognition System

Following the successful results of the phone recognition pilot systems outlined in the previous section, the experimental framework was extended to be able to perform word recognition experiments. As the TIMIT corpus was built for phone recognition experiments rather than word recognition tasks, a new corpus more suitable for evaluating a word recognition task was chosen. The ARPA Continuous Speech Recognition Pilot (WSJ0) corpus [16] was selected as the target corpus for this task. This is a corpus of native English speakers of both genders reading excerpts from the Wall Street Journal. Specifically this work examines the evaluation of the WSJ0 5,000 word vocabulary task. In this task, an evaluation set of 330 utterances from 8 different speakers using a vocabulary limited to 5,000 specific words is performed across all systems. A training set of 7138 utterances from 83 speakers is used to build recognition models, and utterances in the training set may include out-of-vocabulary words for the 5,000 word task. A development set of 368 utterances from 10 speakers is used to tune the models prior to evaluation. All systems are evaluated using the same bigram language model provided with the corpus specifically for the evaluation of the 5,000 word vocabulary task.

As in the phone recognition work described above, the inputs to the CRF models are the outputs of a set of MLP ANNs trained to do frame-level phone classification. However the WSJ0 corpus does not provide phone-level transcriptions for each utterance – only word-level transcripts are provided by the corpus. Frame-level phone class targets for training MLPs and CRF models must be obtained from these word-level transcripts. For these experiments, the HTK toolkit [73] was used to train a standard HMM ASR system using 39-dimensional input vectors of 12 MFCC + energy coefficients along with first and second-order deltas. This system was then used to perform a frame-level Viterbi alignment of the WSJ0 training corpus to provide label targets for both MLP and CRF training.

MLP ANNs were built using the Quicknet MLP framework [12] in the manner described previously in Section 3.1. These MLP networks were trained using a nine-frame window of 12 PLP + energy coefficients along with first and second-order deltas as inputs, with the target labels determined by the frame-level alignment as described above. For MLP training, the training set of the WSJ0 corpus was further divided into a 75 speaker, 6488 utterance MLP training set and an 8 speaker, 650 utterance cross-validation set. MLPs were trained to convergence on the held out development set. The MLPs were constructed with a 4000 hidden unit hidden layer and provide for 54 target output labels.

For these experiments, the best results were obtained using the linear outputs of the MLPs as input features to the CRF rather than the posterior MLP outputs. The models were trained using the stochastic gradient descent training method outlined in Section 2.3.2. The same breakdown of training and cross-validation used for MLP training is used for CRF training, and CRF training stops when the improvement in the phone-level accuracy of the cross-validation set ceases. The CRF models are then used to generate a vector of

local posteriors for each frame of input data. These posteriors are generated for the entire training set as well as for the development and evaluation sets.

Finally, the Crandem models were trained using HTK by using the local posteriors generated by the CRF models as inputs to the HMM. The HMMs were trained over the entire training set of 7138 utterances from all 83 speakers and tuned on the development set of 368 utterances from 10 speakers. These HMMs are all tied-state, triphone HMM systems, with 16 Gaussians per mixture. As described in the previous section, the best results were obtained by using the log transformed posterior outputs of the CRF, with an application of a Karhunen-Loève (KL) transform of the features. Dimensionality was also reduced on all systems after the KL transform and tuned on the development set for performance.[12]

The Crandem system is compared to two other systems as baselines. The first is the standard HMM system built using MFCCs that was used above to generate label files for MLP training. The second baseline is a Tandem HMM system built using the same linear MLP ANN outputs used to train the CRF models. Both Tandem and MFCC-based systems use tied-state, triphone models, with 16 Gaussians per mixture model. Both HMMs were tuned on the same development set as the Crandem model described above to obtain best performance. The only components of these systems that vary are the input feature sets - all other components, including the bigram language model and lexicon, are the same across all systems.

[12]The $unnorm$ transform discussed in regards to the phone recognition experiments above was also examined with this dataset but the performance of this transform was substantially worse than for the log transform. It is suspected that the much longer utterances in the WSJ0 corpus - along with the commensurate much higher normalization term and much larger subsequent values for the $unnorm$-transformed features - are probably to blame for this behavior, but this suspicion remains unconfirmed. All results in this section are reported only on log transformed posteriors.

Model	Training Iterations	Dev WER	Eval WER
MFCC Baseline	NA	9.3%	8.7%
MLP Tandem	NA	9.1%	8.4%
Crandem	1	8.9%	9.4%
Crandem	10	10.2%	10.4%
Crandem	20	10.3%	10.5%

Table 4.4: *WER comparisons across models for development and evaluation sets. Significance at the $p \leq 0.05$ level is at approximately 0.9% percentage difference for each of these data sets.*

4.4 Results & Analysis

Table 4.4 compares the two baseline models to the results of the Crandem system after 1, 10 and 20 iterations of CRF training. Each of the above HMM-based models has 16 Gaussians per mixture. The MLP Tandem model had its best performance on the development set when the 54 dimensional output of the MLP was reduced to 39 dimensions, while the Crandem systems all had their best performance on the development set when the 54 dimensional output of the CRF local posterior calculations were reduced to 21 dimensions.

As the results show, a single iteration of CRF training using the MLP posteriors as inputs produced an statistically insignificant ($p \leq 0.05$) degradation in the WER of the evaluation set over the baseline MFCC system and significant ($p \leq 0.05$) degradation in the WER over the baseline MLP system. Surprisingly, further iterations of CRF training lead to an increase in the error rate rather than a reduction. To check the possibility that the Crandem system is behaving in a radically different manner on WSJ than the previously discussed phone recognition systems trained on TIMIT, phone recognition results were

Model	Training Iterations	Dev Phone Accuracy
MFCC Baseline	NA	70.1%
MLP Tandem	NA	75.6%
Crandem	1	72.8%
Crandem	10	72.8%
Crandem	20	72.9%
CRF	1	69.5%
CRF	10	70.6%
CRF	20	71.0%

Table 4.5: *Phone accuracy comparisons across models for the development set. Significance at the $p \leq 0.05$ level is at approximately 0.6% percentage difference for this data set.*

obtained. Table 4.5 shows the phone accuracy for each of the above systems on the development set, and makes it clear that the degradation of word error rates noted above comes despite a (non-significant) increase in the phone accuracy of the models. Additionally, Table 4.5 shows that as with the phone recognition experiments, the Crandem models show an improvement in phone accuracy over decoding directly off of the CRF itself, though unlike the phone recognition experiments in these experiments the basic MLP Tandem model performs significantly $(p \leq 0.05)$ better than the best Crandem model for phone recognition. This is at least partially due to the fact that during these experiments it was found that tuning the CRF to optimize phone recognition accuracy led to degraded performance for word recognition. As such the results reported here show the results of the Crandem and CRF systems tuned for the best word error rate, not the best phone accuracy.

Is it possible that there is some characteristic of the Crandem-style features that make them behave differently for word recognition than for phone recognition? Figure 4.3 shows an utterance from the development set that compares the initial MLP activation value per

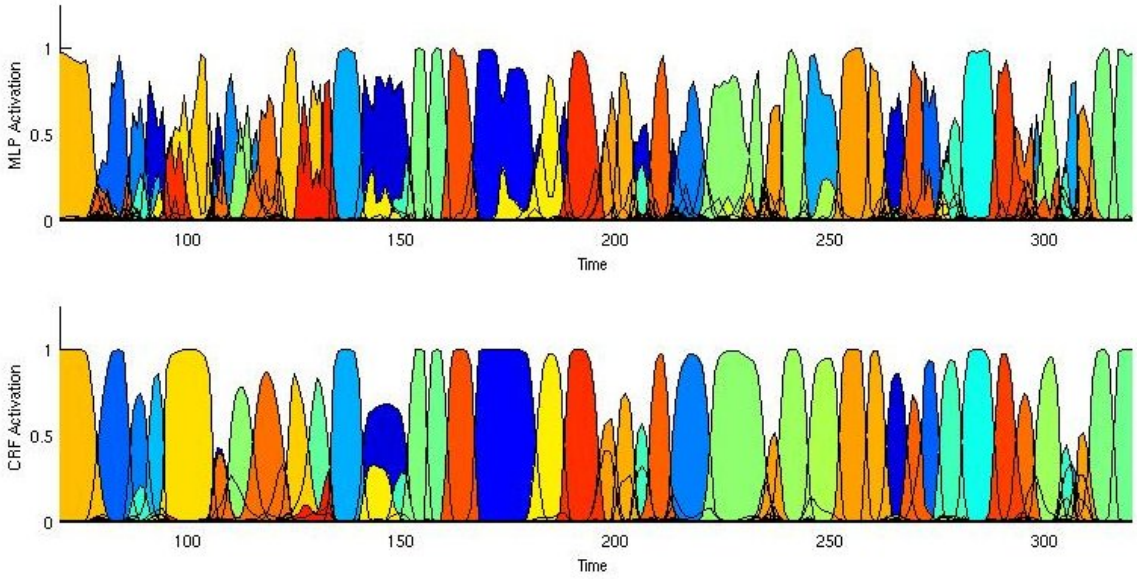

Figure 4.3: MLP activation vs. CRF activation

frame to the activation value per frame of a set of posterior features from a CRF after one training iteration. This example shows that the CRF produces a smoother set of activations than the initial MLP outputs – more of the activations from the CRF produce outputs close to a value of 1.0 and sustain this value over multiple frames of speech. Conversely, the MLP outputs, though smooth in some places, show a much stronger tendency toward jagged peaks – representing areas where the MLP scored a much higher value for a particular phone in a single frame than in surrounding frames. This behavior is observed consistently within the CRF features in the development set as well as within the training set.

The transition features of the CRF model provide an explanation for the smoother graphs of the CRF posterior outputs. In these experiments, only a bias feature is used for each possible transition. However, this single feature is enough to introduce a Markov dependency in the CRF outputs that is not explicitly defined in the MLP outputs. These

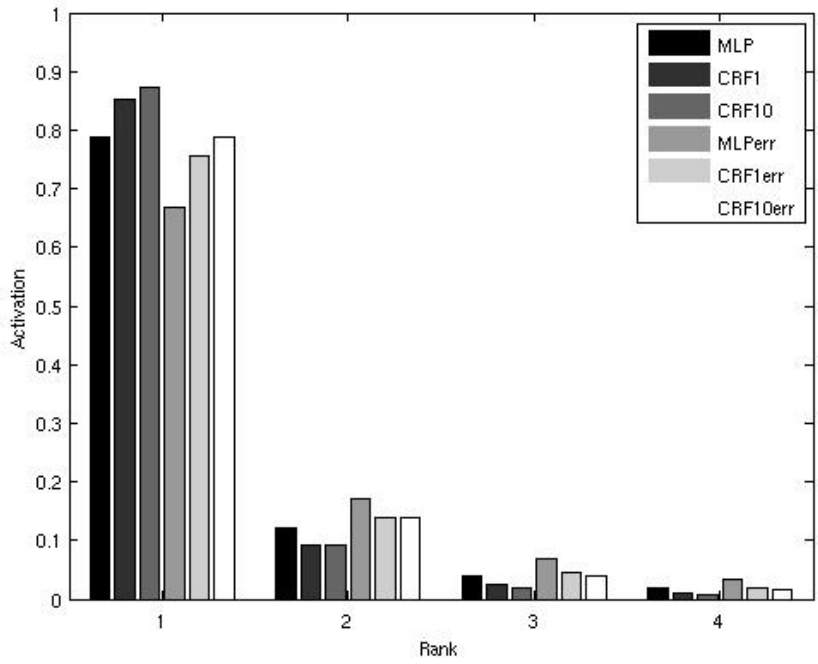

Figure 4.4: Ranked Average Per Frame activation MLP vs. CRF. Activation indicates the value of the model in posterior space. Rank indicates the position of the score descending from the highest score at rank 1 down to the fourth-highest score at rank 4. MLP indicates the average value output by the MLP when the correct class is scored the highest by the MLP at each rank. CRF1 and CRF10 indicate the same score for the CRF trained to 1 iteration and to 10 iterations respectively. MLPerr indicates the average value output by the MLP when the correct class is not scored the highest by the MLP (i.e. when the MLP has made an error) at each rank. CRF1err and CRF10err indicate the same value for the CRF trained to 1 iteration and 10 iterations respectively.

transition features cause the CRF model to prefer a more gradual change in the magnitude of the various phone output values than even the MLP model with a context window of 9 frames produces.

Another factor in the smoothing of the output space for the CRF posteriors is that the CRF on average produces higher values for the phone class with the highest score in a single frame than the MLP classifier on the same frame, pushing the peaks of the scores higher

on the CRF relative to the MLP. Figure 4.4 shows the average value of the top five highest valued classes per frame computed over the development set (results from the training set show a similar pattern). Note that the average score of the top ranked class in each frame is larger for the CRF than for the MLP (0.853 vs 0.788). Conversely, for the lower ranking classes the CRF produces smaller values on average than the MLP (0.092 vs. 0.120 for the average second highest per frame value). This is another factor that leads to the smoothing seen in Figure 4.3 — the value of the highest scoring class is pushed closer to one while the values of the nearest competitors are pushed closer to zero relative to the MLP outputs. This behavior holds for the top 12 classes in the development set (14 in the training set). The lower ranking classes receive values very close to one another between the MLP and the CRF features and very close to zero overall.

Recall that the Crandem system required a much larger dimensionality reduction on the input features than the Tandem system. These smoother outputs help to explain this more extreme dimensionality reduction — the overall space being described by the CRF outputs is much less complex in nature, with reduced variation overall, and so fewer dimensions are needed to perform recognition over this new space. In addition, this smoothing effect may help to explain the degraded performance on word recognition after multiple iterations of CRF training. Figure 4.4 also shows a comparison of the ranked average class values of frames marked as phone errors by the phone recognition process over our development set. The gap between the average value of the top ranked class and the second or lower ranked classes is much larger for the CRF than for the MLP, and gets larger with more iterations of CRF training. This behavior in the features is not surprising – this separation of classes is what is expected from a discriminative model like a CRF. But this behavior suggests a reason for our degraded performance in word recognition. When a phone error is made by

Model	Training Iterations	Eval WER
MFCC Baseline	NA	8.7%
MLP+MFCC Tandem	NA	7.1%
Crandem+MFCC	1	7.1%
Crandem+MLP	1	8.8%

Table 4.6: *WER comparisons with MFCCs on the evaluation set. Significance at the* $p \leq 0.05$ *level is at approximately 0.9% percentage difference for each of these datasets.*

the CRF (i.e. when the highest scoring class is not the correct class), these larger distances between the classes make it harder for the system to fit the observation to the Gaussians for the correct class, making it more difficult for the system to choose between alternatives and leading to a word error. Analysis of the development set suggests that at least in some cases this is likely occurring even between the MLP-Tandem system and single iteration Crandem system, though it does not explain all of the differences in word error between the MLP-Tandem system and the Crandem systems.

Tandem systems are often implemented with both MLP and MFCC features concatenated together as inputs. Table 4.6 compares the results of a Crandem system with MFCC features appended to a similar Tandem system. Here we can see that the MLP-Tandem system and the Crandem system perform comparably, with the difference between the two systems being statistically insignificant and both systems performing significantly ($p \leq 0.005$) better than the baseline system trained only on MFCCs. Table 4.6 also includes a system trained on both the MLP and CRF outputs concatenated together, which performs insignificantly worse on the evaluation set than the MLP-Tandem system shown in Table 4.4, suggesting that the CRF estimates are not providing information that is suitably distinct from the original MLP features.

4.5 Input Feature Transformation

The character of the output of the CRF local posteriors described in Section 4.4 indicates that the CRF model is pushing the values of the local posterior estimators to extremes – far more than that discriminative training done by the MLPs. The gap between the probability assigned to the "best" class label and the competing labels is larger in a CRF than for the comparable MLP. As discussed previously, the results from the initial Crandem experiments suggest that these extreme gaps make the CRF local posterior features a poor fit to Gaussian models (even after log and KL transformations have been applied).

In an attempt to test the hypothesis that the disparity in posterior values shown in Figure 4.4 is to blame for this poor performance, a transformation of the posterior results from the CRF model was examined. This transformation involves taking a root of Equation 4.1, re-normalizing the results over the new possible values, and using these transformed results in place of Equation 4.2 to generate input features for the Crandem system.

The transform is as follows. First, Equation 4.1 is transformed back to the notation used in Equation 2.16:

$$P(\mathbf{y}|\mathbf{x}) = \frac{\exp \lambda \cdot \mathbf{F}(\mathbf{y}, \mathbf{x})}{Z(\mathbf{x})} \tag{4.6}$$

First the transform $R_n(x)$ is defined as follows:

$$R_n(P(\mathbf{y}|\mathbf{x})) = P(\mathbf{y}|\mathbf{x})^{\frac{1}{n}} = (\frac{\exp \lambda \cdot \mathbf{F}(\mathbf{y}, \mathbf{x})}{Z(\mathbf{x})})^{\frac{1}{n}} = \frac{(\exp \lambda \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}))^{\frac{1}{n}}}{Z(\mathbf{x})^{\frac{1}{n}}} \tag{4.7}$$

Next, the transform $T_n(x)$ is defined as a normalization of R_n over all possible label sequences:

$$T_n(P(\mathbf{y}|\mathbf{x})) = \frac{R_n(P(\mathbf{y}|\mathbf{x}))}{\sum_{\mathbf{Y}} R_n(P(\mathbf{Y}|\mathbf{x}))} \tag{4.8}$$

Expanding Equation 4.8 using Equation 4.7 provides:

$$T_n(P(\mathbf{y}|\mathbf{x})) = \frac{\frac{(\exp \lambda \cdot \mathbf{F}(\mathbf{y},\mathbf{x}))^{\frac{1}{n}}}{Z(\mathbf{x})^{\frac{1}{n}}}}{\sum_{\mathbf{Y}} \frac{(\exp \lambda \cdot \mathbf{F}(\mathbf{Y},\mathbf{x}))^{\frac{1}{n}}}{Z(\mathbf{x})^{\frac{1}{n}}}} \tag{4.9}$$

Which simplifies to:

$$T_n(P(\mathbf{y}|\mathbf{x})) = \frac{(\exp \lambda \cdot \mathbf{F}(\mathbf{y},\mathbf{x}))^{\frac{1}{n}}}{\sum_{\mathbf{Y}} (\exp \lambda \cdot \mathbf{F}(\mathbf{Y},\mathbf{x}))^{\frac{1}{n}}} = \frac{\exp \frac{1}{n} \cdot \lambda \cdot \mathbf{F}(\mathbf{y},\mathbf{x})}{\sum_{\mathbf{Y}} \exp \frac{1}{n} \cdot \lambda \cdot \mathbf{F}(\mathbf{Y},\mathbf{x})} \tag{4.10}$$

Note that $T_n(P(\mathbf{y}|\mathbf{x}))$ is in exactly the same form as the equation of the CRF model given in Equation 4.6, except for the addition of the constant term $\frac{1}{n}$. The derivation of the forward-backward algorithm for providing local posteriors in Equation 4.2 still holds. In fact, Equation 4.10 shows that this transform can be considered (and implemented) as a simple transform of the weight vector λ - if each of the elements of λ is simply normalized by n, then Equation 4.10 exactly matches Equation 4.6.

Note also that as long as $n \geq 1$, the relative ordering of the possible sequences $\mathbf{y}$ in Equation 4.10 is the same as for Equation 4.6 as the root function $x^{\frac{1}{n}}$ is monotonic for $x \geq 0$. This means that the transformation T_n will not affect the phone accuracy or correctness of the original CRF model. Only the magnitude of the values output by Equation 4.2 are affected. High scoring values are reduced, while low scoring values are increased by this transformation, resulting in a less extreme divergence in competitor classes.

Table 4.7 shows the results of a system using features transformed by this model. For comparison purposes, the baseline and untransformed Crandem values from Table 4.4 are repeated here. The value of n was determined experimentally on the development set. For

Model	Training Iterations	Dev WER	Eval WER
MFCC Baseline	NA	9.3%	8.7%
MLP Tandem	NA	9.1%	8.4%
Crandem	1	8.9%	9.4%
Crandem (transformed)	1	8.4%	8.5%
Crandem	10	10.2%	10.4%
Crandem (transformed)	10	8.5%	8.8%
Crandem	20	10.3%	10.5%
Crandem (transformed)	20	8.5%	8.5%

Table 4.7: *WER comparisons across transformed models on development and evaluation sets. Significance at the $p \leq 0.05$ level is at approximately 0.9% percentage difference for each of these data sets.*

these experiments, the best results were found when n was close to the magnitude of the λ-weight with the largest absolute value.

The feature transformation has a noticeable effect on the accuracy of the final system. The transformed Crandem features now perform slightly (though insignificantly) better than the MFCC baseline features, rather than insignificantly worse. More tellingly, the transformed features after a single iteration of training now perform almost the same as the MLP Tandem baseline – the difference in these two systems is no longer significant $(p \leq 0.05)$. In addition, although further iterations of CRF training still produce systems that are somewhat worse than the initial system, the degradation is much smaller and the differences in accuracy from one iteration to 20 iterations is not significant with the transformed systems. The transformed systems also required a much smaller degree of dimensionality reduction to be competitive with the MLP and MFCC systems – in the results reported the Crandem (transformed) systems all use the same dimensionality as the MLP Tandem system.

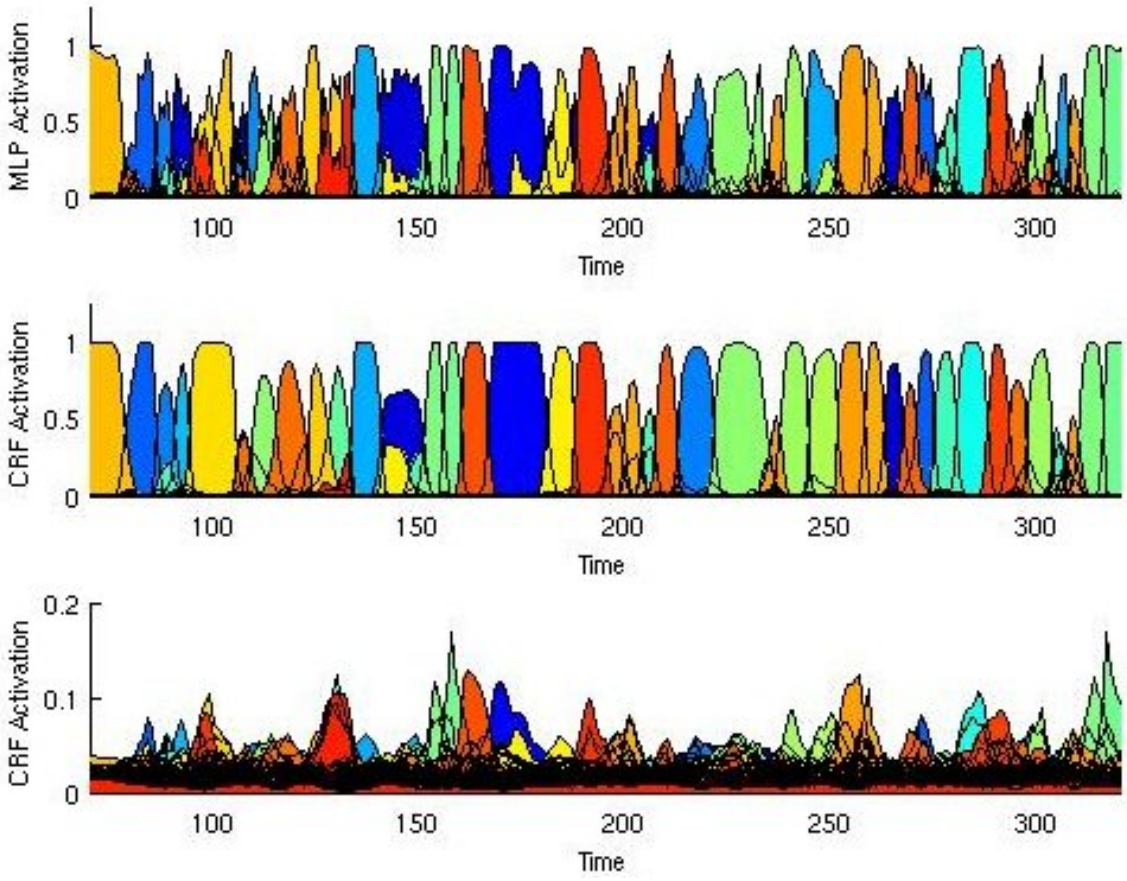

Figure 4.5: MLP activation vs. CRF activation vs. Transformed CRF activation

As a visual example of this effect, Figure 4.5 shows a reprise of Figure 4.3 with the addition of the local outputs created by the transformed CRF for the utterance. Note that the character of the outputs from the transformed CRF are markedly different from both the original MLP and the original CRF. Obviously the overall values output by the transformed CRF are lower than the original CRF outputs, but the shapes of the outputs have also changed substantially. The transitions between frames are less smooth overall and the values of competing classes sit closer to one another. Taken together, the results in Table 4.7

suggest that the extreme divergence in values between the "best" class predicted for a frame by the CRF and the competitor classes is a factor in these systems.

4.6 Summary

These experiments are the first attempt at the use of CRFs for the task of word recognition, and these initial results show a working system for incorporating CRF results into a word recognition system. These results make clear some of the challenges for using features like these in an HMM-based word recognition system. While the Crandem model showed some clear performance gains over the base Tandem system in the task of phone recognition, the results of the word recognition task show areas where the model needs to be carefully considered. Although the local posteriors of the CRF model are superficially like the posteriors produced by an MLP, they are very different in character. Because the CRF model predicts the probability of a label sequence over an entire utterance, rather than by frame-by-frame estimates, the resulting posteriors for frame-level posteriors are higher and smoother from frame to frame – leading to more extreme distinctions within a frame and features that are not as well-modeled by the Gaussian Mixture Model framework of the Tandem HMM.

The mismatch between what CRF models provide the best phone recognition and what CRF models provide the best word recognition is another area that suggests itself for examination. This effect suggests that perhaps there is a mismatch between the goal of maximizing the frame-level accuracy (as the base CRF model attempts to do) and maximizing the word-level accuracy that needs to be improved. Incorporating word-level training criteria into CRF training could lead to improved results.

The system described in this chapter was an attempt to use the CRF model to generate features for an HMM-based ASR system. The main benefit of this approach is that it allows the results of CRF models to be used in mature HMM-based ASR technology systems. In the following chapter a different approach is examined, one where the statistical model for ASR is modified to allow a CRF model to be used as a replacement for an HMM.

CHAPTER 5: WORD RECOGNITION VIA DIRECTLY DECODING FROM CRF MODELS

While the Crandem system outlined in the previous chapter shows one method for incorporating features from a CRF model into a word recognition system, it is not the only method that can be used. The similarities in structure between a CRF and an HMM suggest an more direct alternative to the problem of extracting words from a trained CRF phone model. This chapter examines such a model for word recognition over CRF phone models. Section 5.1 shows how to expand on standard HMM-based ASR systems to derive a direct decoding model for CRF word recognition. Section 5.2 describes a pilot system built using this model, as well as results from this model applied to the TIDIGITS dataset. Finally, section 5.3 describes the results of a larger scale word recognition experiment using the 5,000 word vocabulary task on the WSJ0 dataset.

5.1 A CRF Model of Word Recognition

Recall from Chapter 2 the statistical model for ASR presented again here in Equation 5.1. The goal of statistical ASR is to find the sequence of words $\hat{\mathbf{W}}$ such that the probability of the word sequence given the input speech signal $\mathbf{X}$ is maximized.

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}) \tag{5.1}$$

To accomplish this, the conditional model is first altered via Bayes' Rule to the likelihood model (reproduced here from Equation 2.2).

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}) = \arg\max_{\mathbf{W}} \frac{P(\mathbf{X}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{X})} \tag{5.2}$$

The prior probability of the speech signal $P(\mathbf{X})$ is dropped because it is the same across all possible word sequences, and the final model is expanded to the form reproduced in Equation 5.3 to provide the ability to model speech as a sequence of phone labels rather than as whole word models.

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \arg\max_{\mathbf{\Phi}} P(\mathbf{X}|\mathbf{\Phi})P(\mathbf{\Phi}|\mathbf{W})P(\mathbf{W}) \tag{5.3}$$

Recall that in this formulation, the likelihood $P(\mathbf{X}|\mathbf{\Phi})$ is called the *acoustic model*, the term $P(\mathbf{\Phi}|\mathbf{W})$ is the *dictionary model*, and the prior probability $P(\mathbf{W})$ represents the *language model*. HMM modeling is used in the acoustic model to determine a lattice of phones. The dictionary model is a relatively simple mapping of words to their phonetic sequences, and the language model is usually approximated with an n-gram language model.

This model for ASR must be reviewed if CRF models are to be incorporated into the model. Where the HMM provides the likelihood $P(\mathbf{X}|\mathbf{\Phi})$, the CRF model instead provides the posterior probability $P(\mathbf{\Phi}|\mathbf{X})$. This indicates that changes to the standard model are needed to make the CRF model work for speech recognition – the CRF acoustic model cannot simply be a "drop-in" replacement for the HMM acoustic model.

To make use of the CRF acoustic model, Equation 5.1 must be revisited. Rather than expanding Equation 5.1 into Equation 5.2 via Bayes Rule, Equation 5.1 can be marginalized over the possible phone sequences $\mathbf{\Phi}$ that give the word sequence $\mathbf{W}$:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \sum_{\mathbf{\Phi}} P(\mathbf{W}, \mathbf{\Phi}|\mathbf{X}) \tag{5.4}$$

Which can be expanded to:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \sum_{\mathbf{\Phi}} P(\mathbf{W}|\mathbf{\Phi}, \mathbf{X}) P(\mathbf{\Phi}|\mathbf{X}) \tag{5.5}$$

Next, we make an independence assumption that the word sequences are independent of acoustics when given the phone sequences (this assumption is also made when using HMM models):

If the word sequence $\mathbf{W}$ is assumed to be independent of the acoustics $\mathbf{X}$ when given the phone sequence $\mathbf{\Phi}$ (an assumption that is also made when using an HMM), this can be written as:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \sum_{\mathbf{\Phi}} P(\mathbf{W}|\mathbf{\Phi}) P(\mathbf{\Phi}|\mathbf{X}) \tag{5.6}$$

$P(\mathbf{W}|\mathbf{\Phi})$ is difficult to compute directly – the same sequence of phones can potentially represent two different word sequences. Take the sequence /dh ah k ae t s ae t/ – this sequence could represent the phrase "the cat sat" or the phrase "the cat's at". Since $P(\mathbf{W}|\mathbf{\Phi})$ is trying to compute the probability of the entire word sequence over the entire phone sequence, a method of directly computing this term is non-obvious.

Instead, Bayes Law can be used to rewrite Equation 5.6 as:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \sum_{\mathbf{\Phi}} \frac{P(\mathbf{\Phi}|\mathbf{W}) P(\mathbf{W})}{P(\mathbf{\Phi})} P(\mathbf{\Phi}|\mathbf{X}) \tag{5.7}$$

Applying the Viterbi approximation to Equation 5.7 to look only for the most probable phone assignment for a word sequence assignment gives:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \arg\max_{\phi} \frac{P(\mathbf{\Phi}|\mathbf{X})}{P(\mathbf{\Phi})} P(\mathbf{\Phi}|\mathbf{W}) P(\mathbf{W}) \tag{5.8}$$

Some of the terms in Equation 5.8 are familiar from Equation 5.3 for HMM word recognition. The *language model* $P(\mathbf{W})$ and the *dictionary model* $P(\mathbf{\Phi}|\mathbf{W})$ remain in the new model described by equation 5.8. The *acoustic model* likelihood has been replaced by the posterior likelihood $P(\mathbf{\Phi}|\mathbf{X})$ and a new term $P(\mathbf{\Phi})$ has been introduced. Unlike the prior probability of the acoustics $P(\mathbf{X})$ in Equation 5.2 the value of this new term, hereafter known as the *phone penalty*, cannot be ignored. $\mathbf{X}$ describes the input speech signal, something that is the same for all possible label hypotheses. P($\mathbf{\Phi}$), in contrast, is the prior probability of the hypothesized phone sequence and so must be represented in the final model.

One item of note is that the form of Equation 5.8 bears a similarity to the form of the ANN/Hybrid system [42]. The difference between the Hybrid/ANN approach and the approach outlined here lies in the differences between the CRF classifier and the MLP classifiers used in the Hybrid/ANN system. The CRF estimates a posterior probability over the entire sequence, while the MLPs in the Hybrid/ANN system estimates frame-level posterior probabilities that are then turned into likelihood estimates to be incorporated into an HMM. As such, the normalization term used in a Hybrid/ANN system is a *per frame* normalization term while the normalization term used in the CRF model normalizes over the entire phone sequence. This allows the normalization term of the CRF model discussed above to be a more complex model than a simple class prior over frames (as is used in the Hybrid/ANN model).

As currently implemented the CRF phone models described in previous chapters do not directly compute the quantity $P(\mathbf{\Phi}|\mathbf{X})$. Instead the CRF models provide the related quantity $P(\mathbf{Q}|\mathbf{X})$ where $\mathbf{Q}$ is a frame-level assignment of phone labels. The difference between $\mathbf{Q}$ and $\mathbf{\Phi}$ is subtle but important: there are many different frame level label assignments $\mathbf{Q}$

that map to the same underlying assignment of phone labels $\mathbf{\Phi}$ for any given input signal $\mathbf{X}$. To account for this, the term $P(\mathbf{\Phi}|\mathbf{X})$ can be marginalized over all possible frame label assignments $\mathbf{Q}$:

$$P(\mathbf{\Phi}|\mathbf{X}) = \sum_{\mathbf{Q}} P(\mathbf{\Phi}|\mathbf{Q},\mathbf{X})P(\mathbf{Q}|\mathbf{X}) \tag{5.9}$$

If the phone sequence $\mathbf{\Phi}$ and the acoustic signal $\mathbf{X}$ are assumed to be independent given the frame-level sequence $\mathbf{Q}$[13], this can be simplified to:

$$P(\mathbf{\Phi}|\mathbf{X}) = \sum_{\mathbf{\Phi}} P(\mathbf{\Phi}|\mathbf{Q})P(\mathbf{Q}|\mathbf{X}) \tag{5.10}$$

And Equation 5.8 can then be written as:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \arg\max_{\mathbf{\Phi}} \frac{P(\mathbf{\Phi}|\mathbf{Q})P(\mathbf{Q}|\mathbf{X})}{P(\mathbf{\Phi})} P(\mathbf{\Phi}|\mathbf{W})P(\mathbf{W}) \tag{5.11}$$

As in the derivation of Equation 5.8, Equation 5.11 replaces the summation over all phones with a Viterbi approximation. This final expansion introduces a new term – $P(\mathbf{\Phi}|\mathbf{Q})$, the prior probability of a phone sequence assignment given a frame level assignment. While this term could be modeled in different ways, this work uses a deterministic mapping between frame level assignments $\mathbf{Q}$ and phone sequence assignments $\mathbf{\Phi}$. Each frame level assignment of labels corresponds to exactly one phone sequence assignment.

This assumption of a deterministic mapping is not held by the CRF phone models covered in previous chapters. In these models, each phone is assumed to have a single state. Figure 5.1 shows an ambiguous phone label sequence from a CRF model. The label sequence in this figure could map to the phone sequence /ay/, the sequence /ay ay/ or even

[13]This simplifying assumption is not strictly true, and it is possible that a more complex model connecting phone sequences to acoustics could improve performance of the final system.

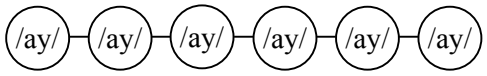

Figure 5.1: Ambiguous single-state CRF model

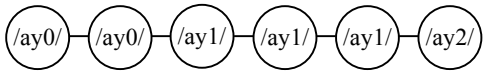

Figure 5.2: Unambiguous 3-state CRF model

the sequence /ay ay ay ay ay ay/. In order to make the map between frame sequences and phone sequences deterministic, a modification to the phone recognition model is needed.

Previous work from HMMs suggests a simple solution to this problem: the use of a multi-state models for single phones [55]. Instead of modeling each phone as a single state, each phone is modeled as a sequence of states requiring at a minimum an entrance state and an exit state, though this work follows typical HMM usage and uses a three state model for phones. Using a three state model the problems cited in the above single state model no longer occur. Figure 5.2 shows an example of this unambiguous 3-state model for a CRF. The sequence in this figure maps only to a single phone: /ay/. The term $P(\mathbf{\Phi}|\mathbf{Q})$ can now be easily implemented as a deterministic map between frame assignments and phone assignments.

5.1.1 CRF Word Recognition Model Implementation

One method for finding the word sequence $\mathbf{W}$ as given by Equation (5.11) is through the use of weighted finite-state transducers (WFSTs). WFSTs have previously been shown to be useful in HMM-based ASR systems [41], and can be modified to be used with the CRF model described above. In this formulation, a finite state transducer is built for each piece of the model (acoustic, dictionary, language) and then composed together using transducer composition. The best path through the composed transducer models then gives a solution for the best word sequence $\mathbf{W}$. Formulating recognition with CRFs in a finite-state transducer composition model allows prior work performed in the area of ASR to be leveraged by the system – the new CRF models can be used where appropriate and proven techniques for ASR can be used where appropriate.

$$A \circ D \circ L \tag{5.12}$$

Equation (5.12) gives a basic structure for this composition for HMM models. Here A is a finite-state transducer that implements the *a*coustic model $P(\mathbf{X}|\mathbf{\Phi})$ from Equation (5.3), D is a transducer that implements the *d*ictionary model $P(\mathbf{\Phi}|\mathbf{X})$, and L is a transducer implementing the *l*anguage model $P(\mathbf{W})$ ($\circ$ is the operator for transducer composition).

In order to use transducer composition, we must select an appropriate semiring to perform operations under. The choice of semiring controls how the weights among arcs between two transducers will be combined in the composition steps. By transforming Equation (5.12) into the log space of the probabilities, either the *log* or *tropical* semirings can be used. These semirings are useful to work with for computational purposes in ASR, as working directly on the probabilities can lead to very small numbers that push the limits of computational precision. Additionally, since the CRF model is an exponential model,

working in the log space allows properties of the model to be exploited that make computation easier, as will be discussed below.

Since word recognition in the CRF model uses a different probabilistic formulation, the transduction model given in Equation (5.12) must be modified to take this into account. Using Equation (5.11) the model becomes:

$$C \circ Q \circ \Phi \circ D \circ L \tag{5.13}$$

where D and L implement the *d*ictionary model and *l*anguage model as in Equation (5.12), but C implements the CRF *a*coustic model $P(\mathbf{Q}|\mathbf{X})$ as given in Equation (5.11), Q is the deterministic mapping $P(\mathbf{\Phi}|\mathbf{Q})$, and Φ implements the *p*hone sequence prior penalty $P(\mathbf{\Phi})$.

Mathematically, the formulation of Equation (5.11) in log space is:

$$\begin{aligned} \hat{\mathbf{W}} = \quad \arg\max_{\mathbf{w}} \arg\max_{\mathbf{q}} \quad & \log P(\mathbf{Q}|\mathbf{X}) - \log P(\mathbf{\Phi}) + \log P(\mathbf{\Phi}|\mathbf{Q}) \\ & + \log P(\mathbf{\Phi}|\mathbf{W}) + \log P(\mathbf{W}) \end{aligned} \tag{5.14}$$

Taking the log of the CRF model given in Equation (2.11) derives:

$$\log P(\mathbf{Q}|\mathbf{X}) = \sum_{t} \lambda \cdot \mathbf{F}(\mathbf{Q}, \mathbf{X}, t) - \log Z(\mathbf{X}) \tag{5.15}$$

where:

$$\lambda \cdot \mathbf{F}(\mathbf{Q}, \mathbf{X}, t) = \sum_{i} \lambda_i f_i(\mathbf{Q}, \mathbf{X}, t) \tag{5.16}$$

In this formulation, the transition weights from one time step to the next are given solely by the combination of the feature functions and associated lambda weights for that particular time step. The combination of these weights is by summation, which is the same

operation that the log and tropical semirings use for combining scores along transducer arcs. Therefore, even though the transition arc weights do not represent probabilities, they are operationally compatible with the log probability weights across the arcs and allow the same operations to be performed. Substituting Equation (5.15) into Equation (5.14):

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{w}} \arg\max_{\mathbf{q}} \begin{array}{l} \sum_t \lambda \cdot \mathbf{F}(\mathbf{Q}, \mathbf{X}, t) - \log Z(\mathbf{X}) \\ - \log P(\mathbf{Q}) + \log P(\mathbf{\Phi}|\mathbf{W}) + \log P(\mathbf{W}) \end{array} \tag{5.17}$$

Additionally, the term $Z(\mathbf{X})$ depends only on X, which is constant for a given input. Thus, this term will have no effect on finding the maximum value for a specific utterance and does not need to be computed, giving instead:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{w}} \arg\max_{\mathbf{q}} \begin{array}{l} \sum_t \lambda \cdot \mathbf{F}(\mathbf{Q}, \mathbf{X}, t) \\ - \log P(\mathbf{\Phi}) + \log P(\mathbf{\Phi}|\mathbf{W}) + \log P(\mathbf{W}) \end{array} \tag{5.18}$$

This formulation gives a model that fits the finite-state transducer paradigm. The language model $P(\mathbf{W})$ and the dictionary model $P(\mathbf{\Phi}|\mathbf{W})$ are built as with standard HMM FST models. $P(\mathbf{\Phi})$ is approximated through the use of a phone-level *n-gram* model trained over the training set of phone transcriptions used to train the CRF model.

Finally a word must be made about some of the practical engineering practices that occur outside of the theoretical framework when implementing an ASR system. In practice, it is often the case that the theoretical model is adjusted through the addition of extra parameters to obtain better control among the interactions of the various models in the system. An example of such a parameter is the language model scaling parameter or language model weight [26]. This parameter is a scaling factor on the language model that provides a method of controlling the interaction of the probabilities between the acoustic model and

the language model. In an HMM-based system such as HTK this parameter is implemented as an exponent on the language model, altering the form of Equation 5.3.

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{w}} \arg\max_{\mathbf{\Phi}} P(\mathbf{X}|\mathbf{\Phi})P(\mathbf{\Phi}|\mathbf{W})P(\mathbf{W})^{l} \tag{5.19}$$

Equation 5.19 provides the altered form of the HMM speech recognition model incorporating a language model weight l into the system. In a similar fashion, the CRF model of ASR can be extended to incorporate a scaling parameter for the language model. As previously discussed, the CRF model of ASR as outlined here has an additional probabilistic model incorporated into it that the HMM speech recognition model does not – the phone penalty model. In order to obtain better control over the interaction of the phone penalty probabilities, the language model probabilities and the acoustic model, the CRF model of ASR was refined to incorporate both a language model scaling parameter l and a phone penalty scaling parameter s. Equation 5.20 shows the form of the CRF model altered to include these parameters.

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \arg\max_{\mathbf{Q}} \frac{P(\mathbf{\Phi}|\mathbf{Q})P(\mathbf{Q}|\mathbf{X})}{P(\mathbf{\Phi})^{s}} P(\mathbf{\Phi}|\mathbf{W})P(\mathbf{W})^{l} \tag{5.20}$$

Equation 5.21 shows the impact of this change for the implementation of the CRF speech recognition model using finite-state transducers. In the log domain, the terms s and l become multipliers for the log probabilities of the phone penalty model and the language model respectively and are easily incorporated into their respective finite-state transducers as scaling parameters.

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \arg\max_{\mathbf{Q}} \begin{array}{l} \sum_t \lambda \cdot \mathbf{F}(\mathbf{Q}, \mathbf{X}, t) \\ -\, s \log P(\mathbf{\Phi}) + \log P(\mathbf{\Phi}|\mathbf{W}) + l \log P(\mathbf{W}) \end{array} \tag{5.21}$$

With a model for CRF speech recognition derived and practical implementation details determined, attention can now turn to evaluation of this model. The sections that follow build upon this discussion to evaluate the CRF model performance in two different tasks – a small vocabulary pilot system over spoken digits and a larger vocabulary system over read speech.

5.2 Pilot System - TIDIGITS

In order to test the model described by Equation (5.18), a pilot system was built over the TIDGITS corpus [37]. The TIDIGITS corpus is a corpus of digits spoken both in isolation and in sequences by male and female, adult and child speakers with diverse selection of dialects from around the United States. The vocabulary of the corpus is restricted to only eleven words – the digits "ZERO" through "NINE" along with the word "OH" which can be used interchangeably with the word "ZERO". For these experiments the child speakers were not used and the training and evaluation sets were restricted to only male and female adult speakers. For this pilot system, the corpus was broken up into 8623 training utterances from 112 speakers, a development set of 847 utterances from 11 speakers and 8700 test utterances from 113 speakers.

For these experiments, new neural networks were not trained over the TIDIGITS dataset. Instead, 39-dimensional PLP coefficients were extracted from the TIDIGITS corpus and a set of MLP neural networks trained on the TIMIT dataset was applied to this data to generate features. This was done primarily because the phone inventory of the TIDIGITS set is quite small in comparison to a larger vocabulary system and the possibility of phone confusions in the feature set is quite low. A set of neural networks trained over TIDIGITS would improve the performance of the system for digit recognition, but would make the lessons

from the system less generalizable to the larger vocabulary case. As such the decision was made to keep the larger phone inventory of the TIMIT dataset despite the mismatch in corpora. The TIMIT neural networks used in this study were constructed with a hidden layer of 2000 units and are the same networks used previously in Chapter 4.

Unlike the TIMIT dataset, the TIDIGITS corpus does not include frame level markings for phone boundaries (or even for word boundaries). Only a word level transcript of each utterance is provided by the dataset. To obtain frame-level label assignments for the TIDIGITS corpus, a CRF model trained on TIMIT MLP outputs was used to force a best-path Viterbi-alignment of labels over the frames of TIDIGITS training data. This best-path label assignment was then used as target assignments for training CRF models for the TIDGITS corpus.

Because the utterances in the TIDIGITS corpus are simply strings of spoken digits, a simple regular expression grammar for digits was implemented in OpenFst and used as the language model for the CRF recognizer. A uniform distribution of phones was also assumed for this pilot system and the phone penalty model was implemented in OpenFst based on this assumption. A small lexicon of word to phone state mappings was also constructed in OpenFst for the digits vocabulary, and these three models were combined together into a single phone penalty/lexicon/language model. At evaluation time, the lattices generated by the CRF model were composed with the combined model and the single best word path extracted from the output.

As a baseline for comparison, a Tandem system was also trained with the same TIMIT-MLP generated TIDIGITS features. As in prior chapters, the system was built using HTK [73]. As with previous Tandem systems, this system was trained using the linear outputs of the MLP transformed via a KL transform. As a second baseline for comparison purposes,

Model	Parameters	Dev. WER	Eval. WER
HTK MFCC Baseline, triphone, 16 Gaussian	˜120,000	0.25%	0.84%
HTK MLP Tandem, triphone, 1 Gaussian	˜7800	1.29%	2.90%
HTK MLP Tandem, triphone, 16 Gaussian	˜120,000	0.57%	1.18%
HTK MLP Tandem, monophone, 1 Gaussian	˜4700	1.26%	3.05%
HTK MLP Tandem, monophone, 16 Gaussian	˜75,000	0.54%	1.28%
CRF (state only), monophone	˜4200	1.11%	2.11%
CRF (state+trans), monophone	˜35,000	0.65%	1.85%
CRF (state only), monophone, 9-windowed	˜34,000	0.67%	1.48%

Table 5.1: *Spoken digit recognition WER comparisons on development and evaluation data sets. Significance at the $p \leq 0.05$ level is at approximately 0.4% and 0.02% respectively.*

a standard HMM system was also trained using 39-dimensional MFCC vectors. For both of these systems, the same training, development and evaluation partitions were used as for the CRF recognizer. Also as with the CRF recognizer, a simple regular expression grammar for digits was used as the language model and one-best evaluation of the words output by the Tandem system was performed to compare to the one-best evaluation provided by the CRF system.

5.2.1 Pilot System Results

The results of the pilot system experiments are shown in Table 5.1. For these experiments, a difference of roughly 0.4% on the development set is significant ($p \leq 0.05$) and a difference of roughly 0.02% on the evaluation set is significant ($p \leq 0.05$). All significance results are considered using a one-tailed Z-test. The CRF system using only state features performs significantly worse than the tied-state triphone Tandem HMM system trained to 16 Gaussians per state or the monophone Tandem HMM system trained to 16 Gaussians

per state. One thing to note is that this CRF system has far fewer parameters than either of these HMM-based systems. To see what effect the reduced parameterization might have, the CRF was also compared to Tandem HMM systems trained only with a single Gaussian per state. The number of parameters for these systems is roughly comparable to the number of parameters for the CRF system. The CRF performs significantly better than either of these two reduced parameter systems on the evaluation set, suggesting that the reduced parameter space may be a problem for the CRF model.

Recalling the experiments from the CRF-HMM systems described in Chapter 4, a CRF was trained with both state and transition feature functions. These results are also shown in Table 5.1. The performance of this system is significantly improved over the performance of a system using only state features, but is still significantly worse than the performance of either of the 16 Gaussian HMM systems on the evaluation set. However it is still of interest that the addition of transition features that merely duplicate the information contained in the state features has caused a significant improvement in the performance of the CRF system.

These results are at least suggestive to the idea that the CRF in this system may not have sufficient parameters for the data it is attempting to model. As another method of increasing the parameterization, a CRF was trained by providing a window of frames as state features rather than a single frame of input. Figure 5.3 shows an example of a CRF where the labels are dependent on a window of input features across time, rather than the features in the frame of time associated with the label. In this figure, the labels of the CRF are dependent on a 3-frame window of input features, with the frame from time t lying a the center of the window of input features for the label for time t. Recall from Chapter 2 that the CRF does not make any assumptions of independence among the input features and that input

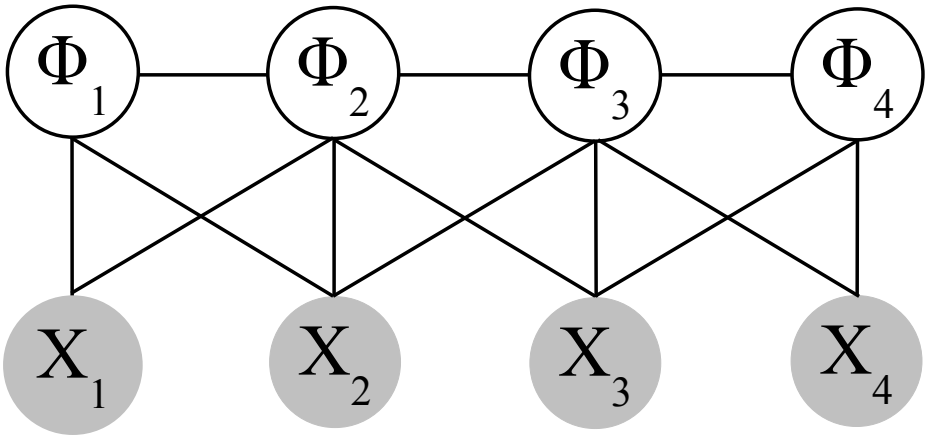

Figure 5.3: Graph of a Linear Chain Conditional Random Field using a 3-frame window of input features

features can be taken from anywhere in the input sequence – this windowed CRF makes use of that property to attempt to provide some context for more accurate recognition.

In this experiment, 9 frames of context were used – 4 frames on either side of the frame at time t for the label at time t. This increases the number of parameters to the CRF to within the same range as for the CRF trained with both state and transition input features. These results are also reported in Table 5.1. These results show a significant improvement over the system using just state features as inputs, but also show a significant improvement over the system using both state and transition features as inputs.

There are two obvious ways that the addition of either the transition features or the window of features could be improving the results of these systems. The fact that these methods increase the number of parameters available to the CRF is discussed above, but additionally these methods both also increase the context available to the CRF. In the case of the window the context is added directly to the evaluation of the current state while in the case of the model using transition features the context is added in the evaluation of the

transitions, but in both cases context information is added to the system. Comparing the results of the Tandem HMM monophone and triphone systems, the triphone system performs slightly but significantly better than the monophone HMM regardless of the number of Gaussians available per state. This suggests that addition of context via the use of triphone labels is a factor in the performance of the HMM systems on this dataset. This lends some evidence to the idea that both the addition of context and the extra parameterization may be influencing the results of the CRF.

Finally, it needs to be noted that neither the Tandem HMM systems nor the CRF systems are able to achieve the same result as the HMM MFCC baseline. Recall that the MLP networks used to generate features for these systems were originally trained on the TIMIT dataset and were not adjusted for the training data in the TIDIGITS dataset. There is a mismatch between the training data for these MLPs and the spoken digits data used in these experiments. This choice was made deliberately. This pilot system was implemented with a plan to expand the system to a larger vocabulary system. While new neural networks could have been trained over the TIDIGITS dataset, the small number of phones in the TIDIGITS lexicon (20 in all) might not have been enough to provide a realistic example of the system working in practice. This situation is rectified in the next set of experiments, where there is no mismatch between the data used to train the MLPs and the data being tested.

5.3 WSJ0 5000 Word Vocabulary Task

Following the evaluation of the pilot system, the CRF system was expanded to accommodate a larger vocabulary task. The target task chosen was the Wall Street Journal 5000 word vocabulary task [16] previously described in greater detail in Chapter 4. As discussed in the previous chapter, the same breakdown of 7138 utterances from 83 speakers is used to

train phone models for recognition for both the CRF and baseline systems. A development set of 368 utterances from 10 speakers is used to to the model, and evaluation is performed over an evaluation set of 330 utterances from 8 speakers. The same bigram language model used in Chapter 4 is used in these experiments.

For this task, the phone class MLP networks used in Chapter 4 were used to train CRFs. As described in the previous chapter, the labels for these classifiers were derived by first training an MFCC-based HMM system and doing a one-best Viterbi forced alignment over the training set. The labels provided by this forced alignment were used as targets for MLP training. For the experiments in this chapter, phonological attribute MLP neural networks of the kind described in Chapter 3 were also required. Targets for neural networks training were acquired using the same process as outlined in Chapter 3 for generating phonological attribute classes over the TIMIT corpus: a canonical set of phonological attribute labels was assigned to each phone label and this association was used to generate phonological attribute labels for each frame based on the phone class label for each frame provided by the forced alignment from the MFCC-trained HMM described above. ICSI Quicknet MLPs [12] were again trained over the extracted 13-dimensional PLPs plus first and second order deltas using these phonological attribute labels as targets. As discussed in Chapter 4, the phone class MLP networks were built using a hidden layer of 4000 units. Due to the reduced label space for the phonological attributes, the phonological attribute classifiers were constructed with a hidden layer of 2000 units.

For multi-state CRF training, phone state labels were required in addition to the phone labels acquired via forced alignment above. These labels were obtained in the same manner as the phone labels for MLP neural network training – the MFCC HMM was used to provide a state-level best-path Viterbi alignment of labels, and these labels were used as

targets for training CRFs. The bigram language model provided for the 5000 word vocabulary task and the lexicon were both implemented as finite-state transducers using OpenFst [58]. The phone penalty model for this task was acquired by using the lexicon to transform the word-level text transcriptions into phone-level transcriptions. An n-gram model was built over the resulting phone transcriptions using the SRI language model toolkit [67], and this n-gram model was then implemented as a finite state transducer and composed with the lexicon and language model transducers to create a final transducer for decoding. Except where stated otherwise, the CRF systems discussed here made use of a bigram phone penalty model.

For efficiency reasons (both memory usage and running time), the best word path for the CRF word recognition model is not found as in the pilot system through direct composition of the acoustic lattice created by the CRF model and the final decoding transducer. This work follows the method of finite state HMM decoding as suggested by Mohri et al in [41] and implements decoding using a time-synchronous, beam-pruning Viterbi decoding algorithm. This allows for a more efficient use of memory, and decoding speeds that are comparable to our baseline HMM implemented in HTK (which is also using a time-synchronous Viterbi decoding method).

The final CRF system is compared to a number of baseline systems. The original MFCC HMM used to derive phone and phone-state labels is shown, as well as an HMM baseline trained on the same PLP coefficients that were used to train the MLP neural networks. In all cases a Tandem HMM system trained using the same input features used by the CRF system is also shown as a baseline. All HMM systems were implemented in HTK, and use the same lexicon and language model used by the competing CRF systems. Unless stated

Model	Dev WER	Eval WER
HTK MFCC Baseline	9.3%	8.7%
HTK PLP Baseline	9.7%	9.8%
HTK Tandem Baseline	9.1%	8.4%
CRF (Phone classes - state only)	11.3%	11.5%

Table 5.2: *Phone class state feature CRF model comparison on development and evaluation sets. Significance at the $p \leq 0.05$ level is at approximately 0.9% percentage difference for each of these data sets.*

otherwise, the HMM systems used in these comparisons were trained as tied-state, triphone context systems with 16 Gaussians per mixture for each state.

5.3.1 WSJ0 5000 Word Vocabulary Task Results

Initial results of the full 5000 word vocabulary task system are shown in Table 5.2. These experiments show a comparison of the HMM baseline to the directly decoded CRF model using input features only as state features. The performance of the CRF system is significantly worse ($p \leq 0.05$) than the performance of any of the baseline HMM systems.

Unlike the CRF system, the HMM systems reported in Table 5.2 are all reported with triphone contexts (as with previous chapters, all CRFs in this section are trained with monophone context on the phone labels). Table 5.3 shows a comparison between the CRF and a monophone HMM system trained on PLP coefficients. The monophone HMM for this task saw its best performance at 32 Gaussians per mixture rather than the 16 Gaussians per mixture used in the triphone HMMs for word recognition. The CRF system substantially and significantly ($p \leq 0.05$) outperforms the monophone HMM system. The fact that there is such a large difference between the monophone and triphone HMM results suggests that

Model	Dev WER	Eval WER
HTK PLP Baseline (triphone)	9.7%	9.8%
HTK PLP Baseline (monophone)	17.5%	16.3%
CRF (Phone classes - state only)	11.3%	11.5%

Table 5.3: *Phone class state feature CRF model comparison (monophones) on development and evaluation sets. Significance at the* $p \leq 0.05$ *level is at approximately 0.9% percentage difference for each of these data sets.*

the lack of context in the CRF system is harming its performance on this dataset. Unlike the TIDIGITS dataset, where the lack of context caused the system to degrade by a small (though significant) amount, the removal of context on this larger vocabulary system caused a large drop in the performance of the HMM-based systems.

One obvious method for adding context to the CRF system would be to use triphone labels rather than monophone labels. Since training paradigm for CRF models performs $O(n^2)$ in the number of labels, an increase from 162 labels to 8748 labels would have a severe negative impact on the training time of the models. Instead this work follows from the work performed previously for the pilot word recognition system outlined in Section 5.2 and adds context via feature functions to the CRF.

For a first attempt at adding context, transition feature functions as described in Chapter 4 were added to the CRF models. The results of these experiments are shown in Table 5.4. With the addition of transition features to the CRF, the performance of the CRF system is insignificantly different from the performance of the Tandem baseline system, despite only operating in the monophone label space rather than the triphone label space of the HMMs. These features seem to have provided some of the benefits of context without the explosion in the size of the label space.

Model	Dev WER	Eval WER
HTK MFCC Baseline	9.3%	8.7%
HTK PLP Baseline	9.7%	9.8%
HTK Tandem Baseline	9.1%	8.4%
CRF (Phone classes - state+transition)	9.2%	8.6%

Table 5.4: *Phone class state + transition features CRF model comparison on development and evaluation sets. Significance at the* $p \leq 0.05$ *level is at approximately 0.9% percentage difference for each of these data sets.*

Recall from Section 5.2 that the addition of transition features, while giving an improvement to the pilot CRF recognition system, did not allow the system to achieve performance similar to the baseline system. With this larger vocabulary system, however, performance compatible to the baseline is achieved. One possible explanation for this may lie in the differences between the two systems use of the *phone penalty* model. Recall that the pilot system was constructed with a uniform, unigram phone penalty model while the systems described above were constructed with a bigram phone penalty model with probabilities derived from the training data. This difference in phone penalty models could provide a source for the relatively larger improvement of this system over the pilot system.

A comparison between systems built with a unigram penalty and a bigram penalty is shown in Table 5.5. Here we can see that the addition of extra context in the phone penalty model has given the bigram system a small improvement over the system implemented with a unigram phone penalty context. While neither of these gains are significant, it does show that the phone penalty model is one source of improvement that the pilot system did not utilize. Systems using more complex trigram and four-gram models were also tested, though these systems performed exactly the same as the bigram system. This suggests that,

Model	Dev WER	Eval WER
CRF (Phone classes - state only - bigram)	11.3%	11.5%
CRF (Phone classes - state only - unigram)	12.3%	12.0%
CRF (Phone classes - state+transition - bigram)	9.2%	8.6%
CRF (Phone classes - state+transition - unigram)	9.6%	8.9%

Table 5.5: *Phone class state features only vs. state + transition features CRF model comparison on development and evaluation sets. Significance at the* $p \leq 0.05$ *level is at approximately 0.9% percentage difference for each of these data sets.*

at least for simple n-gram models of the phone penalty model, while some surrounding context is important to word recognition longer term context is less important than the immediate context surrounding the phone.

Recall that in the previous section performance was improved on the spoken digits system when a window of input features was used to train the CRF rather than just the single current frame of input features. To see if this performance improvement carries through on the larger vocabulary task, a windowed CRF was examined. As with the system in the pilot, this system was trained over a nine-frame window of input features, with the window centered on the current frame. Results of this system are shown in Table 5.6. The system built with windowed state features performs slightly but insignificantly better than the system built using non-windowed features on the evaluation set, and slightly but insignificantly worse on the development set.

One possible explanation for the lack of improvement using windowed features is the larger number of contexts in which phones can appear in this model when compared to the limited vocabulary digits system. In the digits system, there are a very small number of contexts that any given phone can appear in, and so features derived from preceding and

Model	Dev WER	Eval WER
HTK MFCC Baseline	9.3%	8.7%
HTK PLP Baseline	9.7%	9.8%
HTK Tandem Baseline	9.1%	8.4%
CRF (Phone classes - state only)	11.3%	11.5%
CRF (Phone classes - state only - windowed)	11.7%	11.0%

Table 5.6: *Phone class state features only vs. windowed state features CRF model comparison on development and evaluation sets. Significance at the* $p \leq 0.05$ *level is at approximately 0.9% percentage difference for each of these data sets.*

following phones provide direct information about the identity of the current phone. In the larger vocabulary system the contexts are more varied and so the relevance of features from neighboring phone realizations is possibly more limited. It might seem like this is an unlikely explanation, as it would seem to indicate that there should not be an improvement when observations are used as transition features either and yet as previously discussed there is a marked improvement when transition features are used. However it should be kept in mind that the transition features are tied to a particular *transition* identity and not to a particular *phone* identity, while the windowed state features are directly tied to a particular *phone* identity. Context on the neighboring phones might not help with making a good choice for a particular phone in the larger vocabulary model, but it seems reasonable that context on a neighboring phone should help with making a good choice for a particular transition.

Another source of differences between the pilot and the larger vocabulary model lies in the nature of the language model each system uses. The spoken digits system uses a regular expression language model over the words in its vocabulary while the larger vocabulary system uses a richer probabilistic bigram model. The spoken digits system relied almost

entirely on the acoustic evidence to distinguish between words, while the larger vocabulary model is able to make use of a more informative distribution of language along with the acoustic evidence to determine the correct words being spoken. Given that the large vocabulary system relies less heavily on the acoustics alone than the digits system, it does not seem unreasonable to assume that improved acoustics would have a greater impact on the digits system than on the larger vocabulary system. This is not a completely satisfactory explanation, however, since if this were true then one would also not expect to see the large gains provided by the transition features in this system.

Feature Combinations

In the phone recognition experiments on TIMIT outlined in Chapter 3, the CRF model performed significantly better than the HMM model when correlated sets of input features were used. To test whether this improvement carries over to the word recognition level, a CRF was trained to use both phone classifiers as well as phonological attribute classifiers over the WSJ0 corpus. As a baseline for comparison, a Tandem HMM system was also trained with both sets of linear features (phone classes and phonological attribute classes) concatenated together before a KL-transform was applied. Two baseline Tandem results are reported here - the first with all 105 features in the Tandem system, after the application of a KL transform as typically applied to Tandem systems. In addition, since the Tandem HMM system shows improved performance when dimensionality reduction is applied following PCA, a result reporting on a Tandem system reduced to only the top 39 dimensions following the KL transform is also reported.

Results of these experiments are shown in Table 5.7. The performance of the CRF system is significantly ($p \leq 0.05$) better than the performance of the Tandem baseline system using all input features, and insignificantly better than the Tandem baseline system

Model	Dev WER	Eval WER
HTK MFCC Baseline	9.3%	8.7%
HTK PLP Baseline	9.7%	9.8%
HTK Tandem Baseline	10.9%	11.5%
HTK Tandem (top 39)	8.6%	8.1%
CRF (Phone classes + Phono.)	8.3%	8.0%

Table 5.7: *Phone and phonological attribute classes CRF model comparisons on development and evaluation sets. Significance at the* $p \leq 0.05$ *level is at approximately 0.9% percentage difference for each of these data sets.*

using only the first 39 dimensions of the input following the KL transform of the features. This is a clear example of the ability of the CRF to deal with highly correlated streams of input features without explicit decorrelation. The Tandem baseline system using all of the input features sees a drop in performance when compared to the Tandem system using only the phone class features in Table 5.2. The HMM system requires not only the application of a KL transform, but also an explicit dimensionality reduction of the input features to achieve the result that the CRF achieves with no explicit decorrelation and no dimensionality reduction.

5.4 Summary

In this chapter, a statistical framework for CRF word recognition based on current HMM recognition models was derived. This framework is one that is able to make use of the same language models and lexicon models used by HMM systems. This chapter also outlined how the finite-state transducer method of speech recognition first described for use with HMMs can be modified to be used by this CRF framework.

Two sets of experiments examining this framework were outlined. The first, a pilot system over spoken digit sequences, showed that a CRF word recognition system was capable of achieving recognition results comparable to an HMM-based system over the same sets of input features. The second, an extension of the pilot system to the Wall Street Journal 5,000 word recognition task, showed that these results were extendable to a larger vocabulary task. Additionally, the results of this second set of experiments provided a clear example that the CRF word recognition framework was able to make use of the CRF's ability to handle multiple sets of highly correlated features without explicit decorrelation. While the results of the Tandem HMM system and the CRF system were comparable, the CRF achieved these results without external adjusting of the input features. No explicit decorrelation or dimensionality reduction was required for the CRF to match the results of an HMM using explicitly decorrelated and reduced features – the CRF system was able to select which features were important for recognition and which features were not without external intervention.

CHAPTER 6: CONCLUSION

This dissertation has presented an exploration of the use of Conditional Random Field models for Automatic Speech Recognition. Specifically, this work has examined the use of CRF models as a family of discriminative statistical models for integrating highly correlated, possibly redundant features – such as phone classes and phonological features – for ASR. A review of these types of features and current methods for incorporating them into speech recognition systems was provided in Chapter 2 along with a review of the statistical model for ASR, and an introduction to the use of CRF models as an alternative answer to the questions posed by these kinds of features for ASR.

The question posed by this dissertation is whether the CRF model truly is suitable for combining together these diverse and correlated sets of features. An initial answer to this question is explored in Chapter 3, where a phone recognition system based on the CRF model is found to be capable of meeting the performance of a triphone, multiple Gaussian mixture model HMM trained over these highly correlated features. Importantly, the CRF model is capable of meeting this performance with a smaller, monophone-only context to its label space, with far fewer parameters than the GMM-based HMM system, and without explicit decorrelation or dimensionality reduction applied to the input features. When compared to a more comparable HMM system - a monophone label system with a comparable number of parameters - the CRF system achieved superior phone recognition results.

Extending these results to the task of word recognition, however, demonstrates that the answer may not be so simple. An initial attempt to use the CRF model to generate features for HMM-based word recognition – described in Chapter 4 – showed that the system was not able to carry this performance forward into the task of word recognition. Instead, the features generated by the CRF caused a degradation in performance of the HMM system rather than an improvement. A thorough investigation of the model showed that by using the CRF to generate local posteriors rather than a global posterior, the local posteriors generated were overconfident compared to those generated by an MLP and this overconfidence negatively impacted the performance of the final HMM system. To test whether this overconfidence was actually an issue, a method for compensating for this overconfidence by transforming the features from the CRF to something less confident was examined. This transformation yielded improved results, suggesting that the overconfidence of the CRF-derived features was a problem for the combined CRF-HMM system.

However, an alternative method for performing word recognition directly using a CRF model (rather than using it to supply features to an HMM) shows that a system built on the CRF model can perform as well as an HMM system, provided sufficient context is supplied to the CRF. This stands in contrast to the previously described phone recognition experiments, where context did not appear to be a major factor. Specifically the use of transition features derived from the observed data, in addition to state features, seems to be a crucial component for word recognition using these CRF models. These experiments show that the anticipated behavior of the CRF – i.e. the ability to handle redundant, correlated data without the need for explicit decorrelation or dimensionality reduction – can be utilized at the word level in addition to the phone level.

Looking forward, this work opens up a number of avenues for explorations of Conditional Random Fields in statistical ASR. The word recognition system described in Chapter 5 provides a suitable base system for expansion. One area ripe for exploration lies in the model for ASR given by Equation 5.12. Instead of turning this posterior probability into a likelihood model, as was done for this dissertation, more interesting language models that compute this posterior probability over phone sequences directly could be examined. Maximum entropy language models [60] are a family of models that might be particularly suited for use in building this type of system. Such a system would be able to eliminate the use of the need for a phone penalty model altogether, and could possibly be jointly trained with the acoustic model to provide better recognition results. Another possibility follows on the segmental CRF approach of Zweig and Nguyen [76] discussed in Chapter 2 – their segmental CRF model computes this quantity and a modified version of their model could be used in conjunction with the CRF acoustic models in this dissertation for recognition purposes.

Another open question involves the question of feature modeling. This work has used a specific set of features across all systems, but it makes no claims that these are the best sets of features to use in these models. Especially of interest are the transition features – in this work no special features were crafted for the transitions. Instead, the same sets of input features were used for both state features and transition features. It is true that the use of this style of transition feature improved word recognition performance, but it is an open question of whether or not performance might improve even more if features were specifically crafted to look for transition information directly and supply it to the model.

A further area of possible interest involves finding more methods of incorporating wider context into these models. It is clear from the experiments in Chapter 5 that incorporating

context yields benefits, but this work makes no claim that the methods for adding context explored here are the only – or even necessarily the best – methods to use. One possible extension that could yield improvements for this model would be to craft features designed specifically for adding contextual information. Given the results of Chapter 5, these should be implemented as something more complex than just a basic window of features. Rather than just looking to surrounding frames, some thought should be put into trying to find methods of creating features that draw information from surrounding *phones* and incorporating this higher level information into the label decision of the current frame.

One possible approach to this goal of drawing higher level phonetic context into the system may be to explore the use of segments based on phonetic landmarks to generate features for the CRF system. While frame-level phonetic landmark detectors could be used to provide new features similar in nature to the phonological attribute detectors used in this dissertation, segmental approaches based on landmarks (e.g. [19],[11]) could provide a different style of feature that may help incorporate wider context into the model. In this approach, rather than attempting to classify the phonological attributes of each frame, the signal is segmented into pieces based on the detection of a variety of acoustic landmarks marking shifts in phone identity in the signal. One obvious extension would be to examine the CRF as a model for integrating this segmental-level information rather than the frame-level information examined in this work. Alternatively, these segmental level features could also be of use as input features to extend the frame level model previously discussed. A segmental approach such as this would allow the system to incorporate context from the next segment or the previous segment, rather than being limited to the surrounding frames, to allow the system to roughly incorporate context from the previous and succeeding phones. Additionally, a segmental approach lends itself to

A second approach to incorporating higher-level context into the CRF model might be to examine incorporating higher-level linguistic structures as a source of evidence to the model. Previous work in ASR has shown that including estimates of syllable onsets [71], syllable nuclei [3], or entire syllable identities [72] [24] can potentially improve ASR performance. The feature functions of the CRF provide a natural method of incorporating information acquired from syllable-level event detectors into the model, allowing an easy method of incorporating frame-level syllable-based detectors into the CRF model. Alternatively, a syllable-based CRF model could also be examined, though the implications of increasing the size of the label space to account for the larger inventory of syllables relative to the inventory of phones must be taken into account.

Finally, the training criterion of these models also holds interest for future experimentation. Specifically, the training criterion used in this work is a frame-level criterion attempting to maximize the probability of a frame-level labeling sequence given an input signal. However there is a mismatch between this training criterion and the evaluation criteria being used for ASR, since the frame sequence labels are merely a means to the ultimate end of ASR – the sequence of words represented by the speech signal. Using a training criterion that maximizes the probability of the phone-level label sequence, or even the word-level label sequence, rather than the frame-level label sequence could lead to a more accurate recognition model and should be examined.

APPENDIX A: DERIVATION OF PHONOLOGICAL ATTRIBUTES FROM TIMIT PHONE LABELS

The phonological attributes used in this dissertation were constructed using a multi-valued framework derived from the chart of the International Phonetics Alphabet (IPA) as determined by the International Phonetics Association [2]. Divisions in this framework follow earlier work performed by Rajamanohar and Fosler-Lussier [56] [57].

Eight different attribute classes are used in this multi-valued framework, as outlined in Table A.1. Every phone (including silence) is assigned with one of the possible sonority values and one of the possible voicing values. Vowels are additionally assigned values for frontness, height, roundness and tenseness. Consonants are assigned values for manner and place. The "n/a" assignment is given in a class if a phone does not have a value in that class (e.g. stop consonants such as /p/ and the vowel HEIGHT class).

Table A.1: *Phonological attribute classes.*

Class	Output Attributes
SONORITY	vowel, obstruent, sonorant, syllabic, silence
VOICE	voiced, unvoiced, n/a
MANNER	fricative, stop, closure, flap, nasal, approximate, nasalflap, n/a
PLACE	labial, dental, alveolar, palatal, velar, glottal, lateral, rhotic, n/a
HEIGHT	high, mid, low, lowhigh, midhigh, n/a
FRONT	front, back, central, backfront, n/a
ROUND	round, nonround, roundnonround, nonroundround, n/a
TENSE	tense, lax, n/a

The following tables provide a breakdown of the assignment of phonological attribute classes to the TIMIT phoneset.

Table A.2: *Sonority class phonological attribute assignments.*

Value	Phones
obstruent	b, bcl, ch, d, dcl, dh, f, g, gcl, hh, hv, jh, k, kcl, p, pcl, q, s, sh, t, tcl, th, v, z, zh
sonorant	dx, l, m, n, ng, nx, r, w, y
silence	epi, h#, pau
syllabic	axr, el, em, en, eng, er
vowel	aa, ae, ah, ao, aw, ax, ax-h, ay, eh, ey, ih, ix, iy, ow, oy, uh, uw, ux

Table A.3: *Voicing class phonological attribute assignments.*

Value	Phones
voiced	aa, ae, ah, ao, aw, ax, axr, ay, b, bcl, d, dcl, dx, eh, el, em, en, eng, er, ey, g, gcl, hv, ih, ix, iy, jh, l, m, n, ng, nx, ow, oy, r, uh, uw, ux, v, w, y, z, zh
voiceless	ax-h, ch, dh, f, hh, k, kcl, p, pcl, q, s, sh, t, tcl, th
not applicable	epi, h#, pau

Table A.4: *Manner class phonological attribute assignments.*

Value	Phones
fricative	dh, f, hh, hv, s, sh, th, v, z, zh
stop	b, ch, d, g, jh, k, p, q, t
closure	bcl, dcl, gcl, kcl, pcl, tcl
flap	dx
nasal	em, en, eng, m, n, ng
approximate	axr, el, er, l, r, w, y
nasalflap	nx
not applicable	aa, ae, ah, ao, aw, ax, ax-h, ay, eh, epi, ey, h#, ih, ix, iy, ow, oy, pau, uh, uw, ux

Table A.5: *Place class phonological attribute assignments.*

Value	Phones
labial	b, bcl, em, f, m, p, pcl, v, w
dental	dh, th
alveolar	d, dcl, dx, en, n, nx, s, t, tcl, z
palatal	ch, jh, sh, y, zh
velar	eng, g, gcl, k, kcl, ng
glottal	hh, hv, q
lateral	el, l
rhotic	axr, er, r
not applicable	aa, ae, ah, ao, aw, ax, ax-h, ay, eh, epi, ey, h#, ih, ix, iy, ow, oy, pau, uh, uw, ux

Table A.6: *Height class phonological attribute assignments.*

Value	Phones
high	ih, ix, iy, uh, uw, ux
mid	ah, ax, ax-h, eh, ey, ow
low	aa, ae, ao
lowhigh	aw, ay
midhigh	oy
not applicable	axr, b, bcl, ch, d, dcl, dh, dx, el, em, en, eng, epi, er, f, g, gcl, h#, hh, hv, jh, k, kcl, l, m, n, ng, nx, p, pau, pcl, q, r, s, sh, t, tcl, th, v, w, y, z, zh

Table A.7: *Frontness class phonological attribute assignments.*

Value	Phones
front	ae, eh, ey, ih, iy
back	aa, ao, aw, axr, el, er, ow, uh, uw
central	ah, ax, ax-h, ix, ux
backfront	ay, oy
not applicable	b, bcl, ch, d, dcl, dh, dx, em, en, eng, epi, f, g, gcl, h#, hh, hv, jh, k, kcl, l, m, n, ng, nx, p, pau, pcl, q, r, s, sh, t, tcl, th, v, w, y, z, zh

Table A.8: *Roundness class phonological attribute assignments.*

Value	Phones
round	ao, axr, er, ow, uh, uw, ux
nonround	aa, ae, ah, ax, ay, eh, el, ey, ih, ix, iy
roundnonround	oy
nonroundround	aw
not applicable	ax-h, b, bcl, ch, d, dcl, dh, dx, em, en, eng, epi, f, g, gcl, h#, hh, hv, jh, k, kcl, l, m, n, ng, nx, p, pau, pcl, q, r, s, sh, t, tcl, th, v, w, y, z, zh

Table A.9: *Tenseness class phonological attribute assignments.*

Value	Phones
tense	aa, ae, ah, ao, aw, ay, er, ey, iy, ow, oy, uw
lax	ax, ax-h, axr, eh, ih, ix, uh, ux
not applicable	b, bcl, ch, d, dcl, dh, dx, el, em, en, eng, epi, f, g, gcl, h#, hh, hv, jh, k, kcl, l, m, n, ng, nx, p, pau, pcl, q, r, s, sh, t, tcl, th, v, w, y, z, zh

Table A.10: *TIMIT phonological features by phone.*

phone	SONORITY	VOICE	MANNER	PLACE	HEIGHT	FRONT	ROUND	TENSE
aa	VOW	VCD	NA	NA	LOW	BAK	NRND	TEN
ae	VOW	VCD	NA	NA	LOW	FRT	NRND	TEN
ah	VOW	VCD	NA	NA	MID	CEN	NRND	TEN
ao	VOW	VCD	NA	NA	LOW	BAK	RND	TEN
aw	VOW	VCD	NA	NA	LOHI	BAK	NRRD	TEN
ax	VOW	VCD	NA	NA	MID	CEN	NRND	LAX
ax-h	VOW	VLS	NA	NA	MID	CEN	NA	LAX
axr	SYL	VCD	APR	RHO	NA	BAK	RND	LAX
ay	VOW	VCD	NA	NA	LOHI	BKFR	NRND	TEN
b	OBS	VCD	STP	LAB	NA	NA	NA	NA
bcl	OBS	VCD	STCL	LAB	NA	NA	NA	NA
ch	OBS	VLS	STP	PAL	NA	NA	NA	NA
d	OBS	VCD	STP	ALV	NA	NA	NA	NA
dcl	OBS	VCD	STCL	ALV	NA	NA	NA	NA
dh	OBS	VCD	FRI	DEN	NA	NA	NA	NA
dx	SON	VCD	FLP	ALV	NA	NA	NA	NA
eh	VOW	VCD	NA	NA	MID	FRT	NRND	LAX
el	SYL	VCD	APR	LAT	NA	BAK	NRND	NA
em	SYL	VCD	NAS	LAB	NA	NA	NA	NA
en	SYL	VCD	NAS	ALV	NA	NA	NA	NA
eng	SYL	VCD	NAS	VEL	NA	NA	NA	NA
epi	SIL	NA	NA	NA	NA	NA	NA	NA
er	SYL	VCD	APR	RHO	NA	BAK	RND	TEN
ey	VOW	VCD	NA	NA	MID	FRT	NRND	TEN
f	OBS	VLS	FRI	LAB	NA	NA	NA	NA
g	OBS	VCD	STP	VEL	NA	NA	NA	NA
gcl	OBS	VCD	STCL	VEL	NA	NA	NA	NA
h#	SIL	NA	NA	NA	NA	NA	NA	NA
hh	OBS	VLS	FRI	GLT	NA	NA	NA	NA
hv	OBS	VCD	FRI	GLT	NA	NA	NA	NA
ih	VOW	VCD	NA	NA	HI	FRT	NRND	LAX
ix	VOW	VCD	NA	NA	HI	CEN	NRND	LAX
iy	VOW	VCD	NA	NA	HI	FRT	NRND	TEN
jh	OBS	VCD	STP	PAL	NA	NA	NA	NA
k	OBS	VLS	STP	VEL	NA	NA	NA	NA
kcl	OBS	VLS	STCL	VEL	NA	NA	NA	NA
l	SON	VCD	APR	LAT	NA	NA	NA	NA
m	SON	VCD	NAS	LAB	NA	NA	NA	NA
n	SON	VCD	NAS	ALV	NA	NA	NA	NA
ng	SON	VCD	NAS	VEL	NA	NA	NA	NA
nx	SON	VCD	NF	ALV	NA	NA	NA	NA
ow	VOW	VCD	NA	NA	MID	BAK	RND	TEN
oy	VOW	VCD	NA	NA	MDHI	BKFR	RDNR	TEN
p	OBS	VLS	STP	LAB	NA	NA	NA	NA
pau	SIL	NA	NA	NA	NA	NA	NA	NA
pcl	OBS	VLS	STCL	LAB	NA	NA	NA	NA
q	OBS	VLS	STP	GLT	NA	NA	NA	NA

Continued on next page

Table A.10 – *Continued*

phone	SONORITY	VOICE	MANNER	PLACE	HEIGHT	FRONT	ROUND	TENSE
r	SON	VCD	APR	RHO	NA	NA	NA	NA
s	OBS	VLS	FRI	ALV	NA	NA	NA	NA
sh	OBS	VLS	FRI	PAL	NA	NA	NA	NA
t	OBS	VLS	STP	ALV	NA	NA	NA	NA
tcl	OBS	VLS	STCL	ALV	NA	NA	NA	NA
th	OBS	VLS	FRI	DEN	NA	NA	NA	NA
uh	VOW	VCD	NA	NA	HI	BAK	RND	LAX
uw	VOW	VCD	NA	NA	HI	BAK	RND	TEN
ux	VOW	VCD	NA	NA	HI	CEN	RND	LAX
v	OBS	VCD	FRI	LAB	NA	NA	NA	NA
w	SON	VCD	APR	LAB	NA	NA	NA	NA
y	SON	VCD	APR	PAL	NA	NA	NA	NA
z	OBS	VCD	FRI	ALV	NA	NA	NA	NA
zh	OBS	VCD	FRI	PAL	NA	NA	NA	NA

BIBLIOGRAPHY

[1] Yasser Hifny Abdel-Haleem. *Conditional Random Fields for Continuous Speech Recognition*. PhD thesis, University of Sheffield, 2006. 16, 17

[2] International Phonetics Association. *Handbook of the International Phonetics Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, Cambridge, England, 1999. 116

[3] Chris Bartels and Jeff Bilmes. Use of syllable nuclei locations to improve ASR. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 2007. 115

[4] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 192–236, 1974. 21

[5] Shuangyu Chang, Mirjam Wester, and Steven Greenberg. An elitist approach to automatic articulatory-acoustic feature classification for phonetic characterization of spoken language. *Speech Communication*, pages 290–311, 2005. 13

[6] Noam Chomsky and Morris Halle. *The Sound Pattern of English*. Harper and Row, New York, NY, 1968. 8

[7] Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1–8, 2002. 28

[8] Li Deng and Jiping Sun. A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features. *Journal of the Acoustical Society of America*, pages 2702–2719, 1994. 15

[9] Li Deng, Dong Yu, and Alex Acero. Structured speech modeling. *IEEE Transactions on Audio,Speech and Language Processing*, pages 1492–1504, 2006. 57

[10] Richard Duda, Peter Hart, and David Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, NY, 2001. 34, 47

[11] Carol Espy-Wilson, Tarun Pruthi, Amit Juneja, and Om Deshmukh. Landmark-based approach to speech recognition: An alternative to hmms. In *Proceedings of Interspeech*, pages 886–889, 2007. 114

[12] David Johnson et al. ICSI quicknet software package. http://www.icsi.berkeley.edu/Speech/qn.html, 2004. 32, 63, 71, 101

[13] Eric Fosler-Lussier and Jeremy Morris. Crandem systems: Conditional random field acoustic models for hidden markov models. In *Proceedings of the ICASSP*, 2008. 59

[14] Joe Frankel and Simon King. A hybrid ann/dbn approach to articulatory feature recognition. In *Proceedings of Eurospeech*, pages 3045–3048, 2005. 13

[15] Joe Frankel, Mirjam Wester, and Simon King. Articulatory feature recognition using dynamic bayesian networks. In *Proceedings of ICSLP*, 2004. 13

[16] John Garafalo, David Graff, Doug Paul, and David Pallett. CSR-I (WSJ0) complete, 2007. 70, 100

[17] John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, Nancy L. Dahlgren, , and Victor Zue. TIMIT acoustic phonetic continuous speech corpus, 1993. 31

[18] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. MIT Press, Cambridge, MA, 2006. 18

[19] James Glass. A probabilistic framework for segment-based speech recognition. *Computer Speech and Language*, pages 137–152, 2003. 31, 114

[20] Asela Gunawardana, Milind Mahajan, Alex Acero, and John C. Platt. Hidden Conditional Random Fields for phone classification. In *Proceedings of Interspeech*, 2005. 15, 16, 19, 23, 24, 27, 28, 54

[21] Carlos Gussenhoven and Haike Jacobs. *Understanding Phonology*. Oxford University Press, New York, NY, 1998. 9, 10

[22] Andrew K. Halberstadt. *Heterogenous Acoustic Measurements and Multiple Classifiers for Speech Recognition*. PhD thesis, Massachusetts Institute of Technology, 1998. 57

[23] Andrew K. Halberstadt and James R. Glass. Heterogeneous acoustic measurements for phonetic classification. In *Proceedings of Eurospeech*, 1997. 31, 32, 66

[24] Alfred Hauenstein. Using syllables in a hybrid hmm-ann recognition system. In *Proceedings of Eurospeech*, 1997. 115

[25] Hynek Hermansky, Daniel Ellis, and Sangita Sharma. Tandem connectionist feature stream extraction for conventional HMM systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000. 1, 13, 19, 33, 34, 39, 60

[26] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A guide to Theory, Language and System Development*. Prentice Hall PTR, Upper Saddle River, NJ, 2001. 5, 93

[27] Biing-Hwang Juang, Wu Chou, and Chin-Hui Lee. Minimum classification error rate methods for speech recogntion. *IEEE Transactions on Speech and Audio Processing*, pages 257–265, 1997. 18

[28] Amit Juneja. *Speech Recognition Based on Phonetic Features and Acoustic Landmarks*. PhD thesis, University of Maryland, 2004. 13

[29] Simon King and Paul Taylor. Detection of phonological features in continuous speech using neural networks. *Computer Speech and Language*, pages 333–353, 2000. 9, 10, 13

[30] Katrin Kirchhoff. *Robust Speech Recognition Using Articulatory Information*. PhD thesis, University of Bielefeld, 1999. 9, 10, 11, 13

[31] Katrin Kirchhoff. Integrating articulatory features into acoustic models for speech recognition. In *Phonus*, 2000. 11

[32] Peter Ladefoged. *A Course in Phonetics*. Harcourt Brace Jovanovich, New York, NY, 1975. 4

[33] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the ICML*, 2001. 15, 16, 20

[34] Benoit Launay, Oliver Siohan, Arun Surendran, and Chin-Hui Lee. Towards knowledge-based features for HMM based large vocabulary automatic speech recognition. In *ICASSP*, pages 817–820, 2002. 14, 60

[35] Chin-Hui Lee. From knowledge-ignorant to knowledge-rich modeling: A new speech research paradigm for next generation automatic speech recogntion. In *Proceedings of the ICSLP*, 2004. 1

[36] Kai-Fu Lee and Hsiao-Wuen Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, pages 1641–1648, 1989. 35

[37] R. Gary Leonard and George Doddington. TIDIGITS speech corpus. Texas Instruments, Inc., 1993. 95

[38] Karen Livescu, James Glass, and Jeff Bilmes. Hidden feature models for speech recognition using dynamic bayesian networks. In *Eurospeech*, 2003. 15

[39] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning*, pages 49–55, 2002. 23, 27

[40] Florian Metze and Alex Waibel. A flexible stream architecture for asr using articulatory features. In *Proceedings of the ICSLP*, 2002. 13, 14

[41] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recogntition. *Computer Speech and Language*, pages 69–88, 2002. 91, 102

[42] Nelson Morgan and Hervè A. Bourlard. Neural networks for statistical recogntion of continuous speech. *Proceedings of the IEEE*, pages 742–770, 1995. 13, 88

[43] Jeremy Morris and Eric Fosler-Lussier. Combining phonetic attributes using conditional random fields. In *Proceedings of Interspeech*, 2006. 30

[44] Jeremy Morris and Eric Fosler-Lussier. Discriminative phonetic recognition with conditional random fields. In *HLT-NAACL Workshop on Compuationally Hard Problems and Joint Inference*, 2006. 30

[45] Jeremy Morris and Eric Fosler-Lussier. Further experiments with detector-based Conditional Random Fields in phonetic recognition. In *Proceedings of the ICASSP*, 2007. 30

[46] Jeremy Morris and Eric Fosler-Lussier. Conditional random fields for integrating local discriminative classifiers. *IEEE Transactions on Audio, Speech, and Language Processing*, pages 617–628, 2008. 30

[47] Jeremy Morris and Eric Fosler-Lussier. Crandem: Conditional random fields for word recognition. In *Proceedings of Interspeech*, 2009. 59

[48] Arthur Nadas, David Nahamoo, and Michael Picheny. On a model-robust training method for speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, pages 1432–1436, 1988. 18

[49] Mari Ostendorf. Moving beyond the 'beads-on-a-string' model of speech. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 1999. 1

[50] Özgür Cetin, Arthur Kantor, Simon King, Chris Bartels, Mathew Magimai-Doss, Joe Frankel, and Karen Livescu. An articulatory feature-based tandem approach and factored observation modeling. In *Proceedings of the ICASSP*, 2007. 14, 19

[51] Daniel Povey and Brian Kingsbury. Evaluation of proposed modifications to MPE for large scale discriminative training. In *Proceedings of the ICASSP*, 2007. 18

[52] Daniel Povey, Brian Kingsbury, Lidia Mangu, George Saon, Hagen Soltau, and Geoffry Zweig. FMPE: Discriminatively trained features for speech recognition. In *Proceedings of the ICASSP*, 2005. 18

[53] Daniel Povey and P.C. Woodland. Minimum phone error and i-smoothing for improved discriminative training. In *Proceedings of the ICASSP*, 2002. 18

[54] Jose Principe, Neil Euliano, and W. Curt Lefebvre. *Neural and Adaptive Systems*. John Wiley and Sons, Inc., New York, NY, 2000. 34

[55] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, pages 257–286, 1989. 7, 8, 90

[56] Monica Rajamanohar. An evaluation of hierarchical articulatory features. Master's thesis, The Ohio State University, 2005. 116

[57] Monica Rajamanohar and Eric Fosler-Lussier. An evaluation of hierarchical articulatory feature detectors. In *IEEE Automatic Speech Recognition and Understanding Workshop*, 2005. 11, 13, 41, 116

[58] Michael Riley, Cyril Allauzen, and Martin Jansche. Openfst: an open-source, weighted finite-state transducer library and its applications to speech and language. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts*, pages 9–10, Morristown, NJ, USA, 2009. Association for Computational Linguistics. 102

[59] Brian Roark, Murat Saraclar, Michael Collins, and Mark Johnson. Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of the ACL*, pages 48–55, 2004. 28

[60] Roni Rosenfeld. *Adaptive statistical language modeling: A maximum entropy approach*. PhD thesis, Carnegie-Mellon University, 1994. 113

[61] S. Sarawagi. CRF package for java. `http://crf.sourceforge.net/`, 2004. 33

[62] Fei Sha and Fernando Pereira. Shallow parsing with Conditional Random Fields. In *Proceedings of HLT-NAACL*, 2003. 16, 23, 24, 25, 27, 62

[63] Fei Sha and Lawrence K. Saul. Large margin hidden Markov models for automatic speech recognition. In *Proceedings of the NIPS*, 2006. 18

[64] Fei Sha and Lawrence K. Saul. Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models. In *Proceedings of the ICASSP*, 2007. 18

[65] Sabato Marco Siniscalchi, Petr Schwarz, and Chin-Hui Lee. High-accuracy phone recognition by combining high-performance lattice generation and knowledge based rescoring. In *Proceedings of ICASSP*, 2007. 57

[66] James C. Spall. *Introduction to Stochastic Search and Optmization*. John Wiley and Sons, Inc., New York, NY, 2003. 27

[67] Andreas Stolcke. SRILM - an extensible language modeling toolkit. In *International Conference on Spoken Language Technology*, 2002. 102

[68] Sebastian Stüker, Florian Metze, Tanja Schultz, and Alex Waibel. Integrating multilingual articulatory features into speech recognition. In *Proceedings of the 8th Eurospeech Conference on Speech Communication and Technology*, 2003. 12, 13

[69] Sebastian Stüker, Tanja Schultz, Florian Metze, and Alex Waibel. Multilingual articulatory features. In *Proceedings of the ICASSP*, 2003. 13

[70] Jiping Sun and Li Deng. An overlapping-feature based phonological model incorporating linguistic constraints: applications to speech recognition. *Journal of the Acoustical Society of America*, pages 1086–1101, 2002. 15

[71] Su-Lin Wu. *Incorporating Information from Syllable-length Time Scales into Automatic Speech Recognition*. PhD thesis, University of California, Berkeley, 1998. 115

[72] Su-Lin Wu, Brian Kingsbury, Nelson Morgan, and Steve Greenberg. Incorporating information from syllable-length time scales into automatic speech recognition. In *Proceedings of the ICASSP*, 1998. 115

[73] Steve Young, Gunnar Evermann, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book*. Cambridge Unveristy Engineering Department, 2002. http://htk.eng.cam.ac.uk. 14, 35, 64, 71, 96

[74] Dong Yu, Li Deng, and Alex Acero. Hidden conditional random fields with distribution constraints for phone classification. In *Proceedings of Interspeech*, 2009. 16

[75] Qifeng Zhu, Andreas Stolcke, Barry Chen, and Nelson Morgan. Using mlp featres in sri's conversational speech recognition system. In *Proceedings of Interspeech*, 2005. 69

[76] Geoff Zweig and Patrick Nguyen. A segmental crf approach to large vocabulary continuous speech recognition. In *Proceedings of ASRU*, 2009. 17, 113